Topic Modeling as a Community-Detection Problem

Yuanming Tao

Supervisor: A/Prof. Eduardo G. Altmann

THE UNIVERSITY OF SYDNEY

A thesis submitted in partial fulfillment of
the requirements for the degree of
B.Sc (Honours) in Applied Mathematics

October 2019

School of Mathematics and Statistics
Faculty of Science
The University of Sydney
Australia

Acknowledgements

First and foremost, I would like to thank my supervisor A/Prof. Eduardo G. Altmann for his guidance, encouragement, and advice throughout my Honours year as his student. I have been extremely lucky to have a supervisor who cared so much about my project, who responded to my questions promptly, and who provided the resources to help me understand the technicalities. I am very thankful for his organizing the joint meetings weekly with other colleagues. This helped me a lot to get exposure to many other interesting topics. I would also like to thank him for his patient guidance on my thesis writing and for all the presentation rehearsals which he devoted a lot of time and attention to.

I am grateful to Dr. Lamiae Azizi for her useful advice and discussions on the Honours project and the Honours talk. And I would like to thank Jiezhong Wu, Xuanchi Liu, Yilin Ma, Lachlan Burton, and Satoshi Komuro for attending my presentation rehearsals and for providing me a lot of great suggestions. Completing the Honours year would have been all the more difficult were it not for the support provided by Ziqi Wang, my parents and my younger sister. I am indebted to them for their company.

Finally, I must express my gratitude to the Centre for Complex Systems at the University of Sydney and A/Prof. Eduardo Altmann for providing me with the funding to perform a presentation of the Honours project and participate in the international conference on complex system held in Nanyang Technological University at Singapore from 29 Sep. to 4 Oct. in 2019.

Abstract

With the the unprecedented growth of textual datasets, more information is created to an extent where it is infeasible for a person to digest all the available content. This motivates the use of topic modeling to automatically infer the hidden topical structures of a large and unstructured collection of documents. This thesis starts with literature reviews on two existing topic modeling techniques – the traditional topic modeling technique that views the text corpora as a word-document matrix and the network approach that represents the texts as a bipartite network of words and documents. However, these two models are based only on word frequencies. The first objective of this thesis is then to extend the network model to incorporate auxiliary information present in text corpora. Then we investigate whether incorporating further information available about documents can improve their classification. The models were extended in the same network framework. Through experiments and quantitative assessments on the Wikipedia articles, we find that the extended topic models fit better to the data as we utilize more auxiliary information and lead to a better classification of documents.

Contents

Chapter

1

Introduction

Summary

1.1 Topic Modeling and Community Detection: An Overview

1.1.1 Topic Modeling: Motivation and a Concrete Example

We live in the information era. Information can be obtained easily and instantly via the Internet. This has fundamentally changed the dynamic of information acquisition. For example, we can (1) seek views from social media, (2) acquire knowledge by visiting digitalized libraries, and (3) know the world by browsing on-line news. As technology develops, more information is created to an extent where it is infeasible for a person to digest all the available content, let alone extract useful information from it. This motivates the use of computational algorithms to *automatically* organize, search, summarize, and analyze these vast amount of information. To this end, researchers, in the field of *natural language processing* (NLP), have developed *topic models* for discovering the abstract topics that pervade a large and unstructured collection of documents.

Topic models were inspired by the *latent semantic indexing* (LSI) [22], and its probabilistic variant, *probabilistic latent semantic analysis* (pLSA) [23]. Pioneered by Blei et al. [26], the *latent Dirichlet allocation* (LDA) extends the pLSI by employing *Bayesian* inference.

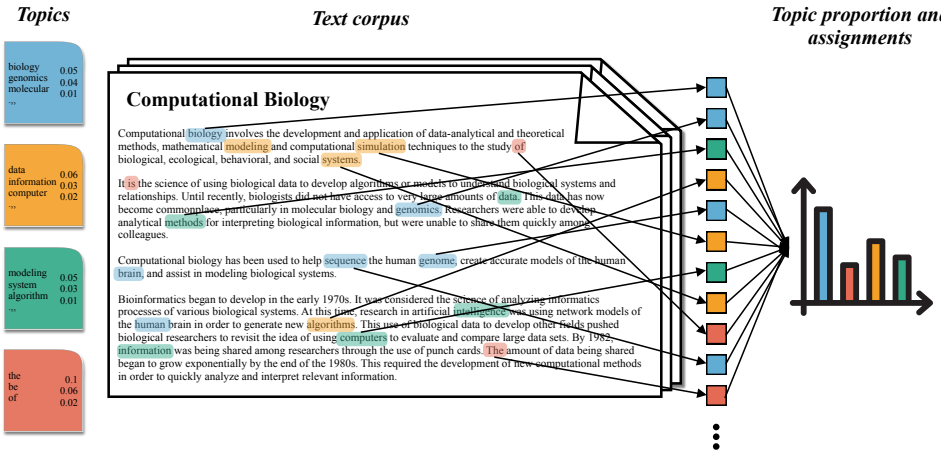


Figure 1.1 – ***The intuitions behind latent Dirichlet allocation.*** *Left: a bunch of corpus-wide topics that describe the thematic structures for the collection of documents. Middle: a collection of documents, called text corpus, with one particular document titled with "computational biology". Right: the topic proportions and assignments of word tokens for the computational biology document. A word token is an instance of a word in a document.*

To illustrate, as shown in Fig. 1.1, the LDA algorithm tries to find some number of "topics" that provides the coarse-grained descriptions for the whole collection of documents. Each topic is a cluster of words with probability distribution over these words. Additionally, the algorithm assumes that each document is a mixture of *corpus-wide* topics and each word token is drawn from one of these topics. It can also help us to find out the proportions of these topics and the assignments of word tokens for any particular document in the text corpus.

1.1.2 A Network Approach to Topic Modeling

More recently, Gerlach, Peixoto, and Altmann obtained a fresh view of the problem of topic modeling by relating it to the problem of finding *communities* in complex networks [28]. They represent the texts as a *bipartite* network of documents and words. The edge between a document and a word in the network represent the number of occurrences of the word in the document. The inferred clusterings of words are the topics. For illustration, we compare the two methods to identify the topical structures in a text corpus composed of 4 documents and 4 distinct words as shown in Fig. 1.2.

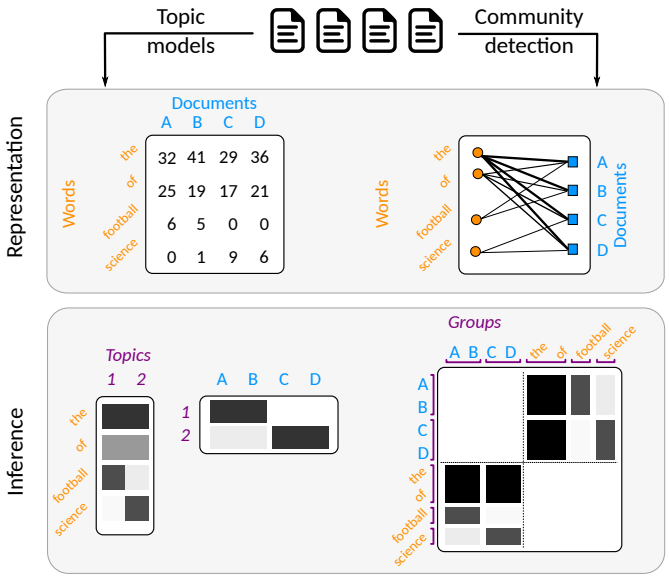
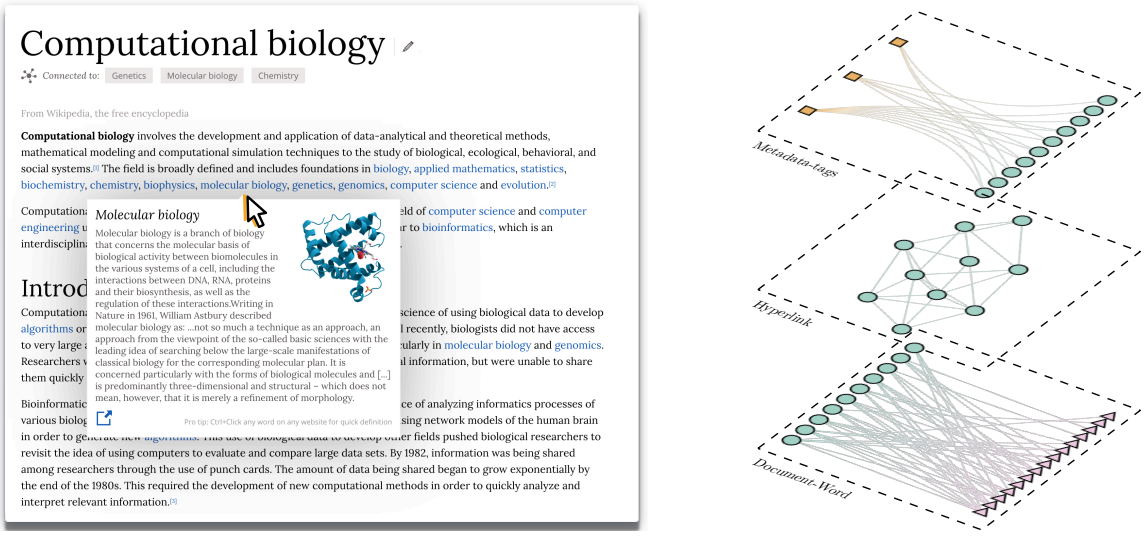


Figure 1.2 – ***Comparison of two approaches to extract topics from collections of texts.*** *The traditional topic modeling views the text corpus as a word-document matrix, where the entries represent word frequencies. And then the matrix is decomposed as a product of two matrices of smaller dimensions with the help of the latent variable topic. The network approach represents texts as a network and infers communities in this network. The nodes consist of documents and words, and edge thickness between them is proportional to the number of occurrences of the word in the document, leading to a bipartite multigraph that is equivalent to the word-document matrix in topic models. Figure adapted from* [28].

They assume that the bipartite network of documents and words is generated by the *stochastic block model* (SBM), which is originally proposed in the social sciences [6], and they also showed its mathematical equivalence to the pLSI. The inference of the communities in such a bipartite network is based on the methods proposed by Peixoto [12].

1.2 Problem Statement and Objectives

The two different approaches to topic modeling discussed above are *only* based on word frequencies. However, in fact, we can have additional information available about the documents. This information can be metadata labels that accompany the text, such as tags, authors and dates; or external links that navigate us to other articles like citations in the context of scientific papers.



(a) – *A Wikipedia article with metadata tags and hyperlinks.*

(b) – *modeling texts with auxiliary information using multilayered network.*

Figure 1.3 – ***Incorporating auxiliary information about documents as additional layers.*** *In additional to document-word bipartite network as the first layer, where the edges represent the word frequencies, we can incorporate metadata tags and hyperlinks of Wikipedia articles as additional layers in a multilayered network.*

For example, on *Wikipedia*, the largest multilingual online encyclopedia, its articles are often associated with several metadata like category labels and the users who have edited the page. Additionally, the Wikipedia article can also contain *hyperlinks* as shown in Fig. 1.3a.

With the additional information about documents available, the question naturally arises of how to account for them when fitting the topic models. This first objective of this thesis is to incorporate the auxiliary information about documents in the same network framework[1]. It is observed that the word tokens, external links between documents, and other metadata labels about the documents are different types of interactions between distinct objects. The main idea is to model them together by a *multilayered network* as illustrated in Fig. 1.3b. The second objective of this thesis is then to investigate whether incorporating further information available about documents can improve their classification.

1.3 Thesis Outline

Based on the above discussions, this thesis is organized as follows. In Chapter 2, we introduce the SBM with its variants that are assumed generative processes for the formation of the document-word bipartite network and the multilayered network considered above. Then, we review the necessary background

[1] In the literature, there exist several extensions for LDA to incorporate metadata about documents. For example, the author-topic model proposed by Rosen-Zvi et al. [24] makes use of authorship information to improve topic modeling. However, in this thesis, we *only* focus on the recent network approach to topic modeling.

for Bayesian inference of the SBMs that will be used throughout. In Chapter 3, we we present the approximation techniques for Bayesian inference of the SBMs. Specifically, we focus on the Metropolis-Hastings algorithm, a variant in the class of Markov chain Monte Carlo (MCMC) methods, for inferring the communities in the networks. In Chapter 4, we move on to descibe the issues of bias and variance trade-off encountered in the Bayesian inference of the SBMs. We then discuss two estimators for the partitions of networks that will be used later in this thesis. For the literature review used in this thesis, we will follow Peixoto's work on Bayesian inference of the SBMs [12, 13, 14, 16, 17, 18, 20].

Next, we show the mathematical equivalence between the pLSI in topic modeling and the SBM in community detection in Chapter 5. And then show how the SBM framework introduced in Chapter 2 can be applied to model texts with/without auxiliary information about documents. Chapter 6 presents our main findings of this thesis. Specifically, we investigate the Wikipedia articles as a case study to show the results of the extended models and how it improves the classification of documents as we incorporate more auxiliary information about the documents.

Finally, Chapter 7 concludes the thesis by summarizing the main results and outline the possible future research directions. To maintain the flow of this thesis, we describe some technical notes in Appendix A.

Chapter 2

Theoretical Framework

Summary

A principled approach to discover the hidden structure of networks is to formulate generative probabilistic models, and then infer their parameters from the observed networks. If the desired structure is composed of modules, a suitable choice for this task will be the stochastic block model (SBM). One of the advantages of the statistical inference approach to community detection over other methods based on heuristics, e.g. modularity method [9] and Louvain methods [8], is that it will not favor partitions that are not backed up by the sufficient statistical significance. Hence, it will not lead us to the spurious partitions in random networks. In this chapter, we review the SBM with its variants, and the inference framework to detect the modular structure of networks that will be used throughout the thesis.

2.1 Classes of Stochastic Block Model

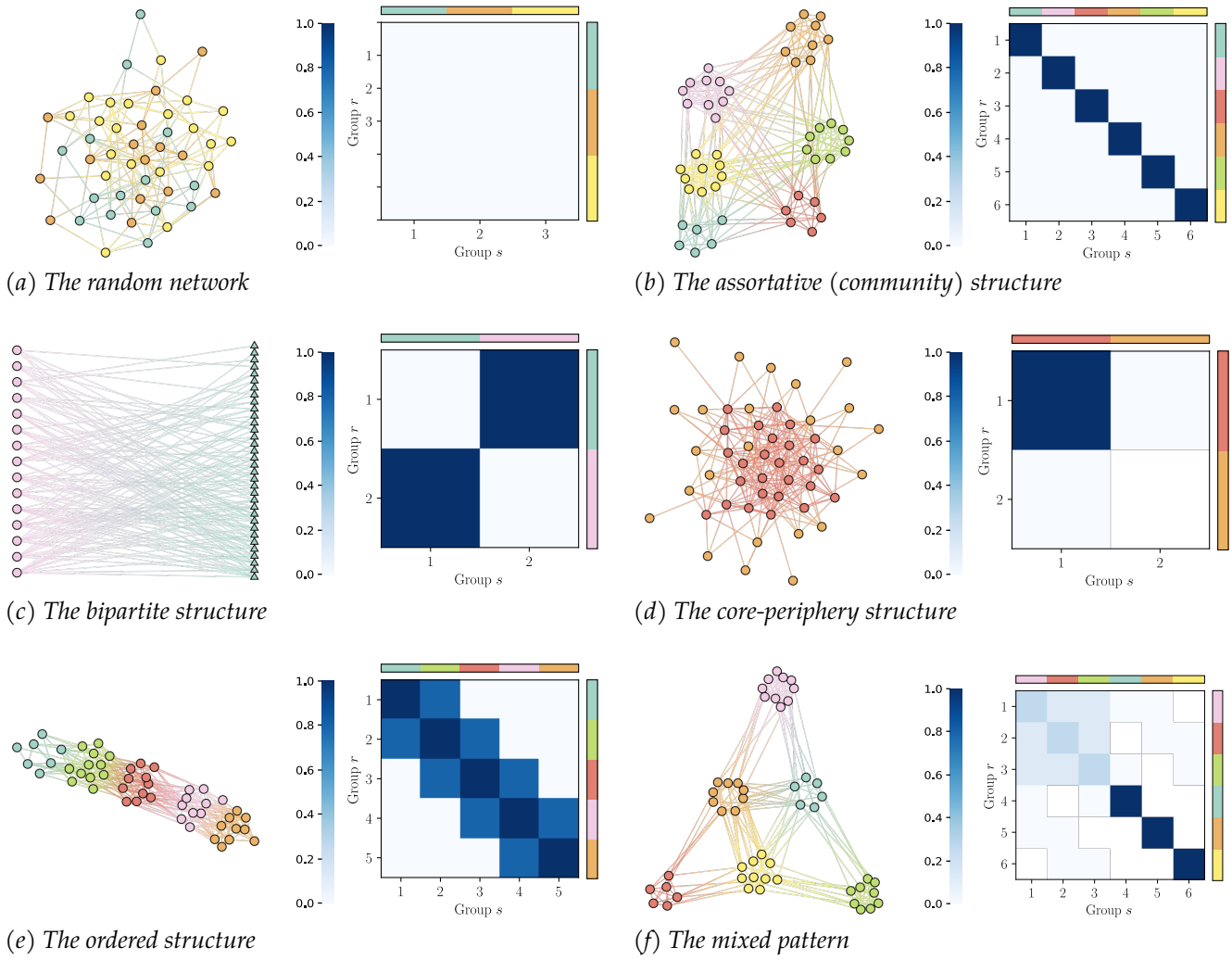
2.1.1 The Standard SBM

Consider a network with adjacency matrix $\boldsymbol{A} = \{A_{ij}\}$ of size N and a partition of nodes $\boldsymbol{b} = \{b_i\}$ into B blocks, where $b_i = 1, \ldots, B$ indicates the block membership of each node. In the simplest form, a stochastic block model assumes that the probability of the existence of a link between nodes i and j depends only on the their block memberships. For simplicity and without loss of generality, in what follows we assume that the networks under consideration are *multigraphs*, i.e. $A_{ij} \in \mathbb{N}$. Assume that the edges are independently sampled from Poisson distributions, the likelihood for an observed network is

$$P(\boldsymbol{A}|\boldsymbol{\lambda}, \boldsymbol{b}) = \prod_{i<j} \frac{\lambda_{b_i b_j}^{A_{ij}} e^{-\lambda_{b_i b_j}}}{A_{ij}!} \tag{2.1}$$

where $\boldsymbol{\lambda} = \{\lambda_{rs}\}$ is the $B \times B$-matrix of group-to-group connectivity rates.

It is emphasized that, while the SBM can perfectly accommodate the usual "community structure" pattern, i.e. there are more links between nodes *inside* the blocks than links *between* the groups and the rest of the network (see Fig. 2.1b), it can equally describe a large variety of other connectivity patterns, such bipartiteness, core-periphery and many others, as illustrated in Fig. 2.1. Hence, the SBM naturally embed the definition of "community" by construction. However, the "community" is not only restricted



(a) *The random network* (b) *The assortative (community) structure*

(c) *The bipartite structure* (d) *The core-periphery structure*

(e) *The ordered structure* (f) *The mixed pattern*

Figure 2.1 – ***The network with planted structure with its corresponding edge probability matrix.*** *The edge probability matrix is visualized as a square matrix with entries for each edge-existence parameter between groups.*

to the community (assortative) structures. In this sense, we instead use the word *modules* to refer to all the connectivity patterns that the SBM can generate.

2.1.2 The Degree-Corrected SBM (DC-SBM)

The standard SBM in the above assumes that nodes that are in the sample group are statistically equivalent. Namely, this implies that all nodes in the same groups have on average the same number of links, which is potentially an unrealistic assumption given that many *real networks* often have very heterogeneous degrees [1]. To allow for degree variability, Karrer and Newman proposed the degree-corrected SBM [7]. Specifically, a new set of parameters called *degree propensity* θ_i of each node i to establish links, so that the likelihood becomes,

$$P(\boldsymbol{A}|\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{b}) = \prod_{i<j} \frac{\left(\theta_i\theta_j\lambda_{b_ib_j}\right)^{A_{ij}} e^{-\theta_i\theta_j\lambda_{b_ib_j}}}{A_{ij}!} \times \prod_i \frac{\left(\theta_i^2\lambda_{b_ib_i}/2\right)^{A_{ij}/2} e^{-\theta_i^2\lambda_{b_ib_i}/2}}{(A_{ii}/2)!}, \tag{2.2}$$

where we also allow for *self-loops*, i.e. an edge that connects a vertex to itself. Within this formulation, θ_i is proportional to i's expected degree and can be different for nodes in the same group, allowing this model to accommodate arbitrary degree sequences within groups. Since this modified model achieves the decoupling the assortative structure from the degree, which are captured separately by the parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$, respectively, the degree variability of the network will not interfere the detection of the communities.

In the degree-corrected SBM, the degrees of the nodes and the number of edges between groups are fixed *only* on average. That is, if we sample networks according to Eqn. (2.2), these quantities can fluctuate between the samples. We refer to such models as the *canonical* formulations of the stochastic block models.

Remarks

It is noted that the above models generates undirected networks. It can be very easily modified to generate directed networks, by making $\boldsymbol{\lambda_{rs}}$ an asymmetric matrix and adjusting the model likelihood accordingly [20].

2.2 Statistical Inference of the DC-SBM

Now that we have the probabilistic generative models to generate artificial networks with prescribed structures, the problem of detecting the modular structure is thus mapped to a problem of statistical inference from the observed networks.

From the frequentists' perspective, a statistical model regards its parameters as *unknown constants*, where the parameters need to be estimated by estimators that are usually obtained from methods such as maximum likelihood estimation (MLE). A disadvantage of classical methods such as MLE is the *overfitting* problem, where if the generative model has a large number of parameters that grows with the observed data, the MLE approach will invariably incorporate a considerate amount of noise [3].

In contrary, a Bayesian model regards the unknown parameters as *random variables*, each of them having a prior distribution of its own. Inference on these parameters are based on their posterior distributions obtained from the Bayes' rule, conditional on the observed data. An advantage of Bayesian inference over the classical approach is that we can incorporate our prior belief of the parameters into the model, where the priors can be based on previous experiences. Even when there is no prior information available, we can let the priors to be "uninformative", and let the data influence the posterior distributions. In this thesis, the approach we adopt to infer modular structures is the Bayesian inference of the stochastic block models, proposed by Peixoto [12].

2.2.1 Nonparametric Bayesian Inference of Partition

Instead of generating networks, we want to determine which partition $\boldsymbol{b}$ generated an observed network $\boldsymbol{A}$, assuming the network is generated by the DC-SBM. By evoking *Bayes' theorem*, we can write the *posterior* distribution of the partition as

$$P(\boldsymbol{b}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b})}{P(\boldsymbol{A})}, \tag{2.3}$$

where

$$P(\boldsymbol{A}|\boldsymbol{b}) = \int P(\boldsymbol{A}|\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{b})P(\boldsymbol{\theta}|\boldsymbol{b})P(\boldsymbol{\lambda}|\boldsymbol{b})\mathrm{d}\boldsymbol{\lambda}\mathrm{d}\boldsymbol{\theta} \tag{2.4}$$

is the *marginal likelihood* integrated over the remaining model parameters, and

$$P(\boldsymbol{A}) = \sum_{\boldsymbol{b}} P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b}) \tag{2.5}$$

is called the *evidence*, i.e. the total probability of the data under the model, which serves as a normalization constant in Eqn. (2.3). In order to compute the marginal likelihood and the posterior of Eqn. (2.3), we need to specify the priors $P(\boldsymbol{\theta}|\boldsymbol{b}), P(\boldsymbol{\lambda}|\boldsymbol{b})$ and $P(\boldsymbol{b})$, which encode our degree of *a priori* belief of the plausibility of the model and its parameters.

Remarks
The model evidence $P(\boldsymbol{A})$ can not be computed exactly since it involves a sum over all possible partitions. However, since it is just a normalization constant, we will not need to determine it when optimizing or sampling from the posterior, as we will see in Chapter 3.

2.2.1.a Determining the Prior Distributions $P(\boldsymbol{\theta}|\boldsymbol{b}), P(\boldsymbol{\lambda}|\boldsymbol{b})$ and $P(\boldsymbol{b})$

The prior distributions are crucial in Bayesian statistics, since they affect the shape of the posterior distributions and thus the inference results. In Bayesian statistical inference, the choice of priors can be determined by past information, such as experiments [2]. However, this is not an applicable scenario when considering networks, where the nodes are unique objects instead of coming from a population. In the absence of such empirical prior information, we should try as much as possible to be guided by reasonable assumptions about the data, rather than *ad hoc* choices. A central proposition we will be using is the *principle of maximum indifference* about the model before we observe any data, which will lead us to *non-informative* prior that assigns equal probabilities to all possibilities conditioned on specific constraints [3].

In this section, we will discuss how to choose the prior distributions for the partitions $P(\boldsymbol{\theta}|\boldsymbol{b})$, the expected degrees $P(\lambda|\boldsymbol{b})$, and the node propensities $P(\boldsymbol{b})$ by obeying this principle.

The Prior for the Partition
We begin by choosing the prior for the partition $\boldsymbol{b}$. The most direct uninformative prior is the flat distribution, where all partitions into at most $B = N$ communities are equally likely, that is

$$P(\boldsymbol{b}) = \frac{1}{\sum_{B=1}^{N} \left\{ {N \atop B} \right\} B!} \tag{2.6}$$

where $\left\{ {N \atop B} \right\}$ are the Stirling number of the second kind that counts the number of ways to partition a set of N elements into B indistinguishable non empty groups and $B!$ recovers the distinguishability of groups. However, it is observed that if N is sufficiently larger than B, there will be much more partitions into $B+1$ groups than there are into B groups. Therefore, the constant prior favors large models with many groups and makes the posterior distribution proportional to the likelihood, which is equivalent to the non-Bayesian approach.

Additionally, we might also want to be agnostic about the number of the groups and we can sample this quantity from its own non-informative distribution $P(B) = 1/N$ and then sample the partition given the number of groups

$$P(\boldsymbol{b}|B) = \frac{1}{\left\{ {N \atop B} \right\} B!} \tag{2.7}$$

since $\left\{ {N \atop B} \right\} B!$ is the number of ways to partition N nodes into B distinct groups.

Upon closer inspection, if we sample partitions from Eqn. (2.7), all group sizes will be approximately the same, which is not a reasonable assumption. Hence, we need a non-informative *hyperprior* on the group sizes $\boldsymbol{n} = \{n_r\}$ where n_r is the number of nodes in group r,

$$P(\boldsymbol{n}|B) = \left(\!\!\binom{B}{N}\!\!\right)^{-1} \tag{2.8}$$

where $\left(\!\binom{n}{m}\!\right) = \binom{n+m-1}{m}$ counts the number of possible histograms that have m counts falling in n bins and is also known as *multiset coefficient*. However, this prior admits the existence of empty groups, which is misleading. For example, if there is a network with 5 communities where one of them is empty, it is equivalent to say the number of communities is 4. Therefore, in order to avoid dealing with empty groups, we can simply exclude them using instead

$$P(\boldsymbol{n}|B) = \binom{N-1}{B-1}^{-1}, \tag{2.9}$$

which is a uniform distribution over all possible histograms that have N counts falling in B non-empty bins.

Conditioned on these randomly sampled sizes, we can then sample the partition using

$$P(\boldsymbol{b}|\boldsymbol{n}) = \frac{\prod_r n_r!}{N!}, \tag{2.10}$$

which is a *maximum entropy distribution*, where all possible partitions are equally likely, given the fixed group sizes.

In summary, we built a hierarchical of priors capturing higher-order aspects of the model. The above gives us finally the hierarchical prior for the partition

$$P(\boldsymbol{b}) = P(\boldsymbol{b}|\boldsymbol{n})P(\boldsymbol{n}|B)P(B) = \frac{\prod_r n_r!}{N!}\binom{N-1}{B-1}^{-1} N^{-1}. \tag{2.11}$$

As above, we went from a naive uninformative prior distribution for the partitions to a Bayesian hierarchy with three levels, where we sample the number of groups, followed by group sizes, and finally the partition. In each of the level, we used maximum entropy distributions constrained on parameters that are sampled from their own distributions from a higher hierarchy. With this, we removed some intrinsic assumptions about the model, i.e. number and sizes of groups, leading to making decision on them until the data is observed. In this sense, the Bayesian approach outlined above is *nonparametric*, where the *order* or *dimension* of the model, specifically, the number of groups B in this case, is the outcome of the inference procedure [12].

The Prior for the Group-to-Group Connections

Another set of parameters for the SBM is the expected degrees between nodes in one group and nodes from different groups. By analogy to the above, we can start with the non-informative prior conditioned on a global average $\bar{\lambda}$, which is the expected density of the observed network. For a continuous random variable x, the maximum entropy distribution with a constrained average $\bar{x}$ is the exponential distribution $P(x) = e^{-x/\bar{x}}/\bar{x}$ [3]. Therefore, for $\boldsymbol{\lambda}$, we have the following

$$P(\boldsymbol{\lambda}|\boldsymbol{b}) = \prod_{r\leq s} e^{-n_r n_s \lambda_{rs}/(1+\delta_{rs})\bar{\lambda}} n_r n_s / \left(1+\boldsymbol{\delta}_{rs}\right)\bar{\lambda} \tag{2.12}$$

with $\bar{\lambda} = 2E/B(B+1)$ determining the expected total number of edges [1].

The Prior for the Node Propensities

It is noticed that the parameters $\boldsymbol{\lambda_{rs}}$ and θ_i always appear in the form of multiple of each other in the likelihood function Eqn. (2.2) of the DC-SBM, we can scale their values arbitrarily in this parametrization $\sum_i \theta_i \delta_{b_i,r} = 1$. Then the interpretation will be: $\boldsymbol{\lambda_{rs}}$ is the expected number of edges between group r and s, $\boldsymbol{\lambda_{rs}} = \langle e_{rs}\rangle$, and θ_i is proportional to the expected degree of node i, $\theta_i = \langle k_i\rangle \,/\, \sum_s \lambda_{b_i,s}$. We can choose the uninformative prior for the node propensities which ascribes the same probability to all possible choices,

$$P(\boldsymbol{\theta}|\boldsymbol{b}) = \prod_r (n_r-1)!\delta\left(\Sigma_i \theta_i \delta_{b_i,r} - 1\right) . \tag{2.13}$$

2.2.1.b Determining the Marginal Likelihood

Combining Eqn. (2.11), Eqn. (2.12), and Eqn. (2.13) and we can compute the integrated marginal likelihood as

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{b}) &= \int P(\boldsymbol{A}|\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{b})P(\boldsymbol{\lambda}|\boldsymbol{b})P(\boldsymbol{\theta}|\boldsymbol{b})\mathrm{d}\boldsymbol{\lambda}\mathrm{d}\boldsymbol{\theta} \\ &= \frac{\bar{\lambda}^E}{(\bar{\lambda}+1)^{E+B(B+1)/2}} \times \frac{\prod_{r<s} e_{rs}! \prod_r e_{rr}!!}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!} \times \prod_i k_i! \times \prod_r \frac{(n_r-1)!}{(e_r+n_r-1)!} \end{aligned} \tag{2.14}$$

where $k_i = \sum_j A_{ij}$ is the degree of node i.

2.2.2 The Equivalence Between the Canonical and Microcanonical Ensembles

The integrated marginal likelihood of Eqn. (2.14) has a special interpretation: it is the joint likelihood of a *microcannonical* model given by

$$P(\boldsymbol{A}|\boldsymbol{b}) = P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b}) \tag{2.15}$$

where

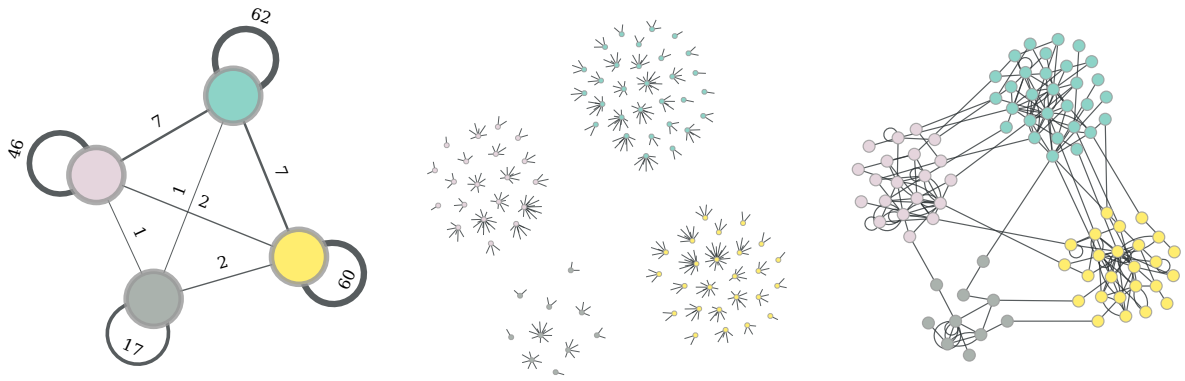
$$P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b}) = \frac{\prod_{r<s} e_{rs}! \prod_r e_{rr}!! \prod_i k_i!}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!! \prod_r e_r!!}, \tag{2.16}$$

[1] It is noted that this uninformative formulation of Eqn. (2.12) also leads to its own problems as with the prior for the node partition. But we postpone the issues to Sec. 2.3.

$$P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b}) = \prod_r \left(\!\!\binom{n_r}{e_r}\!\!\right)^{-1}, \tag{2.17}$$

$$P(\boldsymbol{e}|\boldsymbol{b}) = \frac{\bar{\lambda}^E}{(\bar{\lambda}+1)^{E+B(B+1)/2}} \tag{2.18}$$

and $\boldsymbol{e} = \{e_{rs}\}$ is the matrix of edge counts between groups. As opposed to the "canonical" models introduced previously, the microcanonical models assume that the model parameters correspond to "hard constraints" that are strictly imposed on the ensemble. This implies that there is only one set of parameter choices that is compatible with the network $\boldsymbol{A}$ and the partition $\boldsymbol{b}$ [20]. The generative process of the microcanonical model of SBMs is illustrated in Fig. 2.2. Hence, the generative processes for the parameters in the model can also be formulated via prior distributions. Fig. 2.2 also explains why the approach is fully nonparametric [12].



(a) – *Edge counts,* $P(\boldsymbol{e}|\boldsymbol{b})$. **(b)** – *Degrees,* $P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b})$. **(c)** – *Network,* $P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$.

Figure 2.2 – *Sketch of the generative process of the microcanonical DC-SBM. Given a partition* $\boldsymbol{b}$*, we first sample the edge counts* (a) *between groups, which allows the edge counts to fluctuate between samples. And this is followed by the degrees of the nodes* (b) *and then finally the network* (c)*. Hence, the edge counts* $\boldsymbol{e}$ *between groups and the degree sequence* $\boldsymbol{k}$ *are fixed without any fluctuations between samples. Adapted from Ref.* [20].

Remarks

Additionally, as shown in [20]*, degree sequences generated by Eqn.* (2.17) *result in exponential degree distributions, which are not quite as heterogeneous as what is often encountered in practice. A more refined approach is to increase the Bayesian hierarchy as done for the prior of the node partition, but we postpone the discussion in the Appendix A.1.1.*

2.2.3 The Minimum Description Length Principle (MDL)

With the above microcanonical interpretation, we can view the posterior distribution in Eqn. (2.3) from the perspective of information theory as follows. If a discrete random variable X has probability mass function P_X, the asymptotic amount of information necessary to describe it is $\ln P_X$ by adopting an optimal lossless coding scheme such as Huffman's code is $-\log_2 P(x)$. It is noted that we choose ln

instead of $\log_2$ and thus the unit of measurements will be *nats* rather than *bits*, where the relationship is 1 nat $= 1/\ln(2)$ bits. Hence, we can write the numerator of Eqn. (2.3) as

$$P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b}) = P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e},\boldsymbol{b}) = e^{-\Sigma}, \tag{2.19}$$

where the quantity

$$\Sigma = -\ln P(\boldsymbol{A},\boldsymbol{k},\boldsymbol{e},\boldsymbol{b}) \tag{2.20}$$

$$= \mathcal{S} + \mathcal{L} \tag{2.21}$$

is called *description length* of the data [4, 5] with

$$\mathcal{S} = -\ln P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{k},\boldsymbol{b}) = -E - \sum_k N_k \ln k! - \frac{1}{2}\sum_{rs} e_{rs} \ln\left(\frac{e_{rs}}{e_r e_s}\right) \tag{2.22}$$

being the number of nats required to describe the network if the model parameters are known, and

$$\mathcal{L} = -\ln P(\boldsymbol{k},\boldsymbol{e},\boldsymbol{b}) = Eh\left(\frac{B(B+1)}{2E}\right) + N\ln B - N\sum_k p_k \ln p_k \tag{2.23}$$

being the amount of information required to describe the model parameters, where $h(x) = (1+x)\ln(1+x) - x\ln x$ is the binary entropy function and p_k is the fraction of nodes with degree k. Hence, the optimal network partition that maximizes the posterior distribution is the one that equivalently minimizes the description length.

With this, it is observed that the Bayesian approach outlined for the degree-corrected SBM above prevents *overfitting* problem in the following way: If the number of groups increases, it will decrease $\mathcal{S}$ but simultaneously increase $\mathcal{L}$. Hence, the latter becomes a penalty that disfavors overly complex models.

In this thesis, we will refer the above model as degree-corrected SBM with the Bayesian hierarchy of non-informative priors and call this model as **DC-SBM** for brevity.

2.3 The Hierarchical SBM (hSBM)

Although the above MDL approach is generally protected against overfitting, it is still susceptible to *underfitting*, i.e. when we mistake statistically significant structure for randomness, resulting in the inference of an overly simplistic model. This happens whenever there is a large discrepancy between our prior assumptions and what is observed in the data. If we revisit the the uninformative prior for $P(\boldsymbol{\lambda}|\boldsymbol{b})$ in Eqn. (2.12), it put approximately equal weight on all allowed types of large-scale structures. As argued before, this seems reasonable at first, since we should not bias our model before we observe the data. However, the implication of this choice is that we expect *a priori* the structure of the network at the aggregate group level, i.e. considering only the groups and the edges between them (not the individual nodes), to be fully random.

In addition, the above DC-SBM with non-informative priors have an optimal number of inferred blocks that scales as $O(\sqrt{N})$. That is, it fails to recognize modules if their sizes are smaller than a scale that depends on the total number of nodes in the network, i.e. smaller blocks tend to be merged together with neighboring groups. This boundary is known as the *resolution limit* [30].

The solution to address the above issues, as proposed by Peixoto [16], is the hierarchical variants of the SBM. The result of an inference of a SBM can be represented by a multigraph where the nodes are the groups and edges are the corresponding edges of the nodes inside each block. The idea of the hierarchical SBM is that the multigraph is again generated by a SBM and we can perform this step recursively until the trivial partition is obtained as displayed in Fig. 2.3.

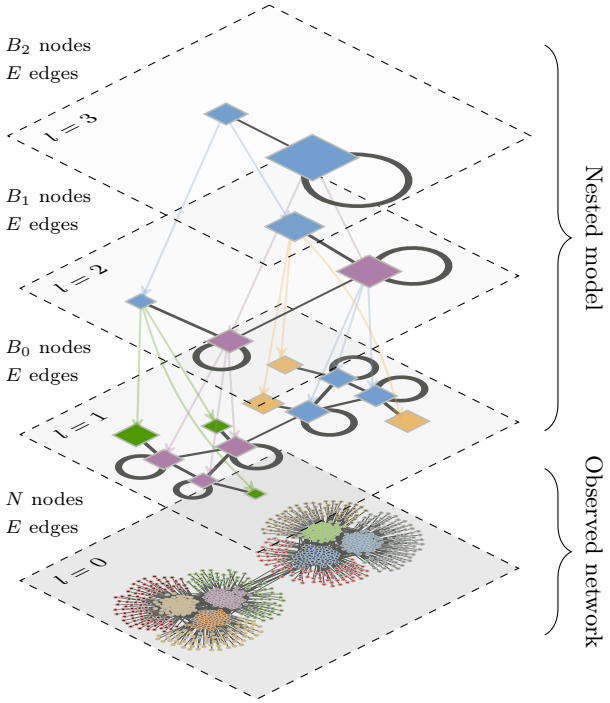


Figure 2.3 – *Sketch of the hierarchical SBM with three levels. A generated network is plotted at the bottom and the top-level structure describes a core-periphery structure, which is then subdivided in the lower levels. Figure adapted from [16]*

More precisely, the hierarchical SBM replaces the uninformative prior in Eqn. (2.18) by a nested sequence of SBMs, and the prior distribution for the matrix of edge counts $\boldsymbol{e}_l$ at level $l \in \{0, \dots, L\}$ is

$$P(\boldsymbol{e}_l|\boldsymbol{b}_{l-1}, \boldsymbol{e}_{l+1}, \boldsymbol{b}_l) = \prod_{r<s} \left(\!\!\binom{n_r^l n_s^l}{e_{rs}^{l+1}}\!\!\right)^{-1} \prod_r \left(\!\!\binom{n_r^l (n_r^l+1)/2}{e_{rs}^{l+1}/2}\!\!\right)^{-1}, \tag{2.24}$$

where B_l, $\boldsymbol{b}_l$ denote the number of groups and the partition at level l, respectively.

The prior for the node partitions is again given by Eqn. (2.11),

$$P(\boldsymbol{b}_l) = \frac{\prod_r n_r^l!}{B_{l-1}!} \binom{B_{l-1}-1}{B_l-1}^{-1} B_{l-1}^{-1}. \tag{2.25}$$

with $B_{-1} = N$. Hence, the joint probability of the observed network, edge counts, and the hierarchical partition $\{\boldsymbol{b}_l\}$ becomes

$$P(\boldsymbol{A}, \{\boldsymbol{e}_l\}, \{\boldsymbol{b}_l\}|L) = P(\boldsymbol{A}|\boldsymbol{e}_1, \boldsymbol{b}_0) P(\boldsymbol{b}_0) \prod_{l=1}^{L} P(\boldsymbol{e}_l|\boldsymbol{b}_{l-1}, \boldsymbol{e}_{l+1}, \boldsymbol{b}_l) P(\boldsymbol{b}_l) \tag{2.26}$$

where the boundary conditions are imposed by $B_L = 1$ and $P(\boldsymbol{b}_L) = 1$.

The inference of the hierarchical SBM is done in the same manner as the noninformarive one, by obtaining the posterior distribution of the hierarchical partition

$$P(\boldsymbol{b}_l) = \frac{\prod_r n_r^l!}{B_{l-1}!}\binom{B_{l-1}-1}{B_l-1}^{-1} B_{l-1}^{-1}, \tag{2.27}$$

and by analogy the description length is given by

$$\Sigma = -\ln P(\boldsymbol{A}|\{\boldsymbol{e}_l\},\{\boldsymbol{b}_l\}) - \ln P(\{\boldsymbol{e}_l\},\{\boldsymbol{b}_l\}) \ . \tag{2.28}$$

In summary, the hierarchical SBM is a degree-corrected SBM which infers a hierarchy of nested SBMs based on the statistical evidence of the given data in a completely non-parametric way, i.e. there are no free parameters to choose beforehand. In what follows, we refer to this model as **hSBM** for brevity.

2.4 The SBM with Independent Layers

As mentioned in Chapter 1, word frequencies that create multiple edges between a document and one word, hyperlinks between documents, and metadata tags between user-generated labels and documents are independent and different types of interactions. These distinct types of interactions can be modeled as layers of networks. The SBM can be generalized to model the generation of these kinds of network structures as well [19]. We thus devote this section to formulate generative models of layered networks in the same SBM framework.

We consider graphs that have a layered structure, so that the adjacency matrix in layer $l \in [1, C]$ can be written as A_{ij}^l, corresponding to the presence of an edge between vertices i and j in layer l, where we assume that $A_{ij}^l \in \mathbb{N}$. It is assumed that the nodes are globally indexed, and a node can receive edges in all layers in principle. The *collapsed graph* corresponds to the merging of all edges in a single layer, with a resulting adjacency matrix $A_{ij} = \sum_l A_{ij}^l$. In what follows, a specific layered graph is denoted as $\{G_l\}$ (with $G_l = \{A_{ij}^l\}$ being an individual layer), and the corresponding collapsed graph as $G_c = \{A_{ij}\}$.

There are two important observations in the generative process. Firstly, we generate each layer as an independent SBM, constrained only by the fact that the group memberships of the nodes are the same across all layers. Furthermore, nodes are only allowed to belong to a subset of the layers, by including a $N \times C$ layer membership matrix $\{z_{il}\}$, where each binary entry $z_{il} \in [0, 1]$ determines whether node i belongs to layer l. That is, if a node is not present in a given layer, it is forbidden to receive edges of that type. We depict this generative process in Fig. 2.4.

Using the shorthand $\{\{\theta\}_l\} = \{\{e_{rs}^l\}\}$ and $\{\phi\} = \{b_i\}$, the likelihood of the resulting layered block model is then

$$P(\{G_l\}|\{\{\theta\}_l\},\{\phi\},\{z_{il}\}) = \prod_l P(G_l|\{\theta\}_l,\{\phi\}), \tag{2.29}$$

with $P(G_l|\{\theta\}_l,\{\phi\})$ being the likelihood of the standard stochastic block model as discussed previously, where G_l is the subgraph containing only the edges of layer l and the nodes specified by $\{z_{il}\}$.

There remains one important modification of the above model, i.e. the degree-corrected assumption as mentioned in Sec. 2.1.2. We incorporate this important aspect by specifying the *layer-specific* degree sequence $\{k_i^l\}$, where $k_i^l = \sum_j A_{ij}^l$ is the degree of node i in layer l, so that $\{\{\theta\}_l\} = \{\{e_{rs}^l\},\{k_i^l\}\}$. Hence,

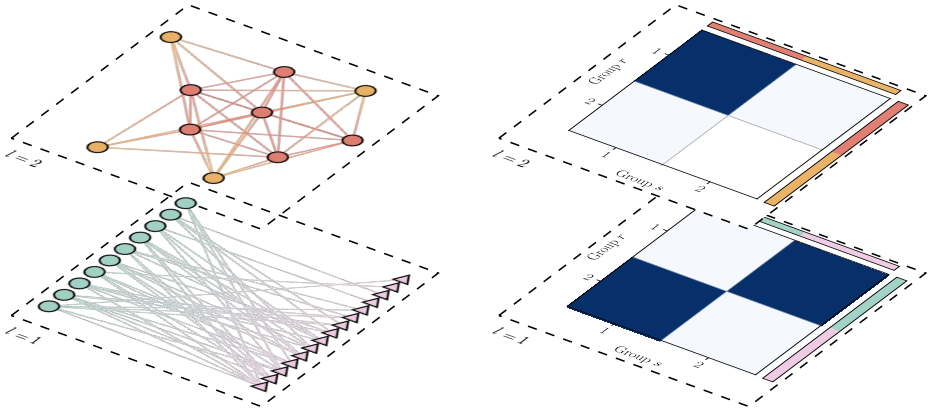


Figure 2.4 – ***SBM with independent layers.*** *Left: The generated networks. Right: The model parameters specified for each layer. In this example, the SBM with independent layers possess different large-scale structures in each layer, where the first layer $l = 1$ is the bipartite structure and the second is the core-periphery structure. The important assumptions are: (1) The layers are formed independently from each other; (2) The degree variability is different across different layers.*

it is important to notice that this model allows for degree variability across different layers, i.e. a node that receives many edges in one layer may possess low degree in another. It is also noted that given the layer-specific degree sequence, we do not need to the layer-membership matrix parameter in the standard SBM case discussed above, since we can set the layer-specific degree of a node to zero so that this node will inherently not receive any edge in that layer. Thus, the layer-specific degree sequence parameters $\{k_i^l\}$ can replace the layer-membership matrix $\{z_{ij}\}$, which can be removed from Eq. 2.29 in this case.

Chapter 3

Inference Algorithm Using Markov Chain Monte Carlo (MCMC)

Summary

As described in Chapter 2, we can write the posterior distributions explicitly for the SBM and its variants, but they are complex to characterize via analytic examination. We can not sample from or maximize the posterior distributions in a direct manner. Indeed, it is also noticed the space we are exploring is tremendous: As the number of nodes increase in a network, the number of all possible partitions follows the Bell number, which grows much faster than the exponential function, as illustrated in Fig. 3.1.

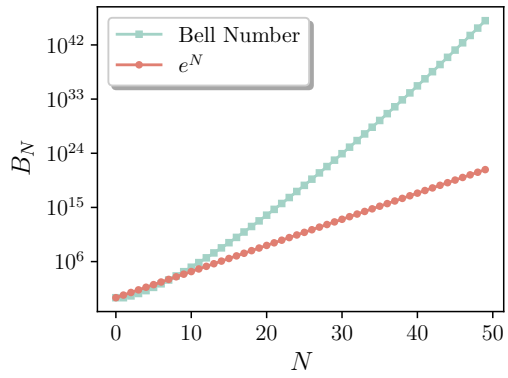


Figure 3.1 – *The number of partitions B_N of a network into communities grows faster than exponentially with the network size N.*

What we can do, however, is to adopt the Markov Chain Monte Carlo (MCMC) methods. Given the posterior distribution as our *target distribution*, the central idea of MCMC algorithm is to construct a Markov chain such that its equilibrium distribution is the posterior distribution we want to sample from or maximize.

The chapter will be organized as follows. For the standard SBM with non-informative prior introduced in Chapter 2, we first discuss its MCMC algorithm in detail. For other variants of the model, we describe them in a summarized manner. Specifically, in Sec. 3.1, we first discuss how to construct a Markov chain by employing the Metropolis-Hastings algorithm [10, 11]. In particular, since we want the chain to take time as less as possible to be equilibrated, we also discuss how to choose a smart proposal distribution to make the algorithm more efficient. We then discuss how to find the maximum of the posterior distribution via *simulated annealing*, so that we can obtain the estimate that maximizes the posterior distribution. Next, although an equilibrium probability distribution (the posterior distribution) that the MCMC converges to is irrespective of the initial starting state, the efficiency and convergence rate of the MCMC algorithm still heavily depends on where the chain starts its exploration. This will be our main focus in Sec. 3.3.

3.1 Sampling From the Posterior

The essential idea to sample from the posterior distribution $P(\boldsymbol{b}|\boldsymbol{A})$ is to start with some arbitrary state $\boldsymbol{b}_0$ and make move proposals $\boldsymbol{b} \to \boldsymbol{b}'$ with a probability $P(\boldsymbol{b}'|\boldsymbol{b})$, such that the equilibrium distribution will be exactly $P(\boldsymbol{b}|\boldsymbol{A})$ after a sufficiently long time.

This process is guaranteed by constructing the Markov Chain that satisfies the following two conditions

1. *Ergodicity*: Every configuration is reachable from any other configurations with non-vanishing probability;
2. *Detailed balance*: The moves are *reversible* and each observed partition must occur with probability proportional to the proposal distribution [1].

Given any arbitrary proposal distribution, with the only condition that it satisfies the first condition, the desired posterior distribution can be guaranteed to be reached eventually by employing the Metropolis-Hasting algorithm [10, 11], indicating that we should accept the move $\boldsymbol{b} \to \boldsymbol{b}'$ according to the probability a given by

$$a = \min\left(1, \frac{P\left(\boldsymbol{b}'|\boldsymbol{A}\right)}{P(\boldsymbol{b}|\boldsymbol{A})} \frac{P\left(\boldsymbol{b}|\boldsymbol{b}'\right)}{P\left(\boldsymbol{b}'|\boldsymbol{b}\right)}\right), \tag{3.1}$$

otherwise the attempted move is rejected. The ratio $P\left(\boldsymbol{b}|\boldsymbol{b}'\right)/P\left(\boldsymbol{b}'|\boldsymbol{b}\right)$ enforces the reversibility property $T\left(\boldsymbol{b}'|\boldsymbol{b}\right)P(\boldsymbol{b}|\boldsymbol{A}) = T\left(\boldsymbol{b}|\boldsymbol{b}'\right)P\left(\boldsymbol{b}'|\boldsymbol{A}\right)$, where $T\left(\boldsymbol{b}'|\boldsymbol{b}\right)$ is the final transition probabilities after incorporating the acceptance criterion of Eqn. (3.1). Additionally, it is important to notice that when computing the ratio $P\left(\boldsymbol{b}'|\boldsymbol{A}\right)/P(\boldsymbol{b}|\boldsymbol{A})$ in Eqn. (3.1), we do not need to determine the normalization constant $P(\boldsymbol{A})$ appeared in Eqn. (2.3) since it cancels out. And hence a can be determined exactly.

3.1.1 A Naive Approach: The Random Move Proposal

The simplest proposal that satisfies the above the two conditions is the fully random move proposal, i.e. to attempt to move each vertex into one of the B blocks with equal probability. However, this proposal can be very inefficient. Specifically, if the network have well-defined structures so that the node will belong to very few of the B blocks with a non-zero probability, most random vertex moves will then be rejected.

Additionally, despite the theoretical guarantees of the Metropolis-Hasting algorithm, a naive implementation of the algorithm may perform very badly. This is because it might take a very long time for the asymptotic properties of the Markov chain to be realized. And thus the desired equilibrium distribution is never reached in practical time. Hence, we devote next section to discuss the smart move proposal proposed by Peixoto [13].

3.1.2 A Smart Move Proposal: Use the Currently-Inferred Structure

Given a node i with block membership s, the more efficient move proposal is defined as

$$p(r \to s|t) = \frac{e_{ts} + \epsilon}{e_t + \epsilon B}, \tag{3.2}$$

[1] Specifically, the probability of the observed partition is proportional to its description lengths.

where t is the group label of a *randomly* chosen neighbor for node i and ϵ is a free parameter to enforce the enforces ergodicity condition[2].

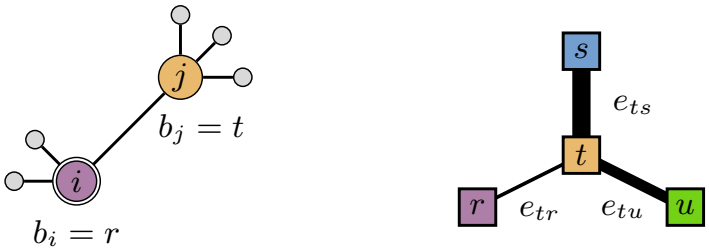


Figure 3.2 – ***Illustration of more efficient move proposal for a node.*** *The move proposal is made by inspecting the neighborhood of node i and randomly selecting one of neighbors j. Based on its group membership $t = b_j$, the edge counts between groups are inspected (right), and the move proposal $b_i = s$ is made with probability proportional to e_{ts}. In this example, the probability of the attempted move $b_i \to s$ is larger than either $b_i \to r$ (no movement) or $b_i \to u$, since $e_{ts} > e_{tr}$ and $e_{ts} > e_{tu}$. Figure adapted from [13].*

What Eqn. (3.2) means, as illustrated in Fig. 3.2, is essentially that the probability move for a node from group r to s is proportional to the number of edge counts e_{ts} between the group s and t. That is, instead of moving a node blindly, we utilize the currently inferred model parameters to choose the most likely blocks to which the original node belongs.

The detailed balance condition can then be enforced using the Metropolis-Hastings criterion by the following acceptance probability a

$$a = \min\left\{e^{-\Delta\Sigma}\frac{\sum_t p_t^i p(s \to r|t)}{\sum_t p_t^i p(r \to s|t)}, 1\right\}, \tag{3.3}$$

where the $e^{-\Delta\Sigma}$ is the difference in the total description lengths between the attempted state and the current state, p_s^i is the fraction of neighbors of node i which belong to block s, $p(t \to r|s)$ is computed after the attempted move $r \to s$, whereas $p(r \to s|t)$ is computed before the attempted move.

Remarks

It is emphasized that the move proposals of Eqn. (3.2) do not *bias the partitions toward any specific kind of mixing pattern, as depicted in Fig. 2.1. This is because they inspect the neighbors of a node only to access with other groups their kinds are typically connected — which can be different from the the group assignment of the original node.*

3.2 Simulated Annealing For Global Maximization of the Posterior

Instead of sampling from the posterior distribution, we want to find the partition that maximizes of the posterior distribution. In this case, we can introduce an "inverse temperature" parameter β in the Eqn. (3.3). So, the final acceptance probability reads

$$a = \min\left\{e^{-\beta\Delta\Sigma}\frac{\sum_t p_t^i p(s \to r|t)}{\sum_t p_t^i p(r \to s|t)}, 1\right\}. \tag{3.4}$$

[2]If we make $\epsilon \to \infty$, we can recover the random move proposal.

The so called inverse temperature β controls the likelihood of negative moves and can be used for *simulated annealing* [35]. Simulated annealing increases the value of β step by step to increase the chance to stay in a local optimum at the end, but leave local optima in the beginning. This can be done by changing the value of β either slowly or abruptly after the chain has been sufficiently equilibrated. In this thesis, we will consider the latter approach to improve the efficiency of the MCMC algorithms.

3.3 Efficient Inference: The Agglomerative Heuristics

Since the convergence of the MCMC algorithm depends heavily on the initial states, we devote this section to discuss a more efficient approach to obtain a starting state that lies close to the mode of the posterior.

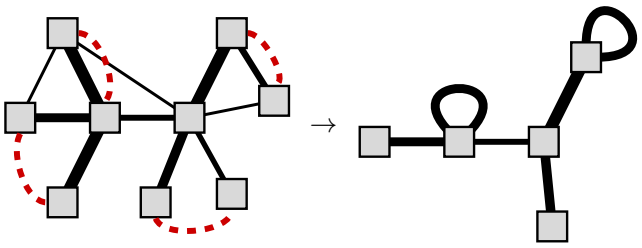


Figure 3.3 – *The initial state of the MCMC can be obtained with an agglomerative heuristic, where groups are merged together using the same proposals described in Fig. 3.2. Figure adapted from* [13].

The agglomerative heuristic approach presented in [13] is to perform one-dimensional optimization (Fibonacci search [36]) on the number of group B, where for each value we obtain the best partition from a larger partition with $B' > B$.

The approach is composed of the following steps taken alternatively: (1) We attempt the moves of Eqn. (3.2) until no improvement to the description lengths is observed. (2) We merge groups together, achieving a smaller number of groups. Specifically, at first bracketing the minimum of Σ by finding a triplet (B_1, B_2, B_3) with $B_1 < B_2 < B_3$ such that $\Sigma|_{B=B_1} > \Sigma|_{B=B_2} < \Sigma|_{B=B_3}$. We can start with $B_1 = 1, B_3 = B_{\max}$ and choosing $B_2 = B_3 - \lfloor B_3 - B_1 \rfloor_F$, where $\lfloor x \rfloor_F$ is the largest Fibonacci number x. This is repeated until the minimum is bracketed. After this, the intervals are progressed bisected with $B_2' = B_3' - \lfloor B_3' - B_1' \rfloor_F$, where (B_1', B_3') is the largest of the intervals (B_1, B_2) or (B_2, B_3). That is, we progressively merge groups together, achieving a smaller number of groups. This step is done by treating each group as a single node and using Eqn. (3.2) as a merge proposal, and selecting the ones that least increase the total description lengths Σ (see Fig. 3.3).

3.4 The MCMC Algorithms for Other SBM Variants

We devote this section describe the MCMC algorithms for the hierarchical SBM introduced in Sec. 2.3. The basic idea in this case is that we proceed in each step of the Markov chain by randomly choosing a level l, and performing the proposals of Eqn. (3.2) on that level, as described in [20].

In this scenario, the posterior distribution of the hierarchical partition is

$$P(\{\boldsymbol{b}_l\}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}, \{\boldsymbol{b}_l\})}{P(\boldsymbol{A})}, \tag{3.5}$$

and this posterior can be factorized as

$$P(\{\boldsymbol{b}_l\}|\boldsymbol{A}) = \frac{\prod_l P(\boldsymbol{e}_{l-1}, \boldsymbol{b}_l|\boldsymbol{e}_l)}{P(\boldsymbol{A})} = \prod_l P(\boldsymbol{b}_l|\boldsymbol{e}_{l-1}, \boldsymbol{e}_l) \tag{3.6}$$

with per-level posteriors

$$P(\boldsymbol{b}_l|\boldsymbol{e}_l, \boldsymbol{e}_{l+1}) = \frac{P(\boldsymbol{e}_l|\boldsymbol{e}_{l+1}, \boldsymbol{b}_l)P(\boldsymbol{b}_l)}{P(\boldsymbol{e}_l|\boldsymbol{e}_{l+1})}, \tag{3.7}$$

where we assume $\boldsymbol{e}_0 = A$, and $P(\boldsymbol{e}_l|\boldsymbol{e}_{l+1})$ is a normalization constant.

Hence, a workable approach is to separately sample partitions at each level according to its individual posterior, conditioned on the remaining levels, which are kept unchanged for the time being. If we sample from each level in this manner we can guarantee ergodicity, and if the moves at the individual levels are reversible, the overall distribution will correspond to the desired full posterior of Eq. 3.5.

Since the hierarchical levels are coupled, when moving a node at level l, we must ensure that this does not invalidate the partition at level $l+1$. Therefore, we must forbid node moves between groups that are themselves at different groups in the next level[3].

In more detail, we proceed as follows. At each individual level l, we perform a move proposal of node i from its current group r to a new group s, according to a probability $P(b_i^{(l)} = r \to s)$ that we will specify shortly.

We compute the difference in the log-likelihood $\Delta \ln P_l$ at that level, and we accept the move according to the Metropolis-Hastings criterion, i.e. with a probability

$$a = \min\left\{1, e^{\Delta \ln P_l} \frac{P(b_i^{(l)} = s \to r)}{P(b_i^{(l)} = r \to s)}\right\}, \tag{3.8}$$

where $P(b_i^{(l)} = s \to r)$ is the probability of the reverse move being proposed. The log-likelihood difference is computed as

$$\Delta \ln P_l = \ln \frac{P(b_i^{(l)} = s, \boldsymbol{b}_l \setminus b_i^{(l)}|\boldsymbol{e}_l, \boldsymbol{e}_{l+1})}{P(b_i^{(l)} = r, \boldsymbol{b}_l \setminus b_i^{(l)}|\boldsymbol{e}_l, \boldsymbol{e}_{l+1})}, \tag{3.9}$$

where $\boldsymbol{b}_l \setminus b_i^{(l)}$ means the partition of the remaining nodes excluding node i.

In this case, instead of the random move proposal that leads to a long *mixing* time, a better approach is to again inspect the current parameters of the model to provide a better guess of the posterior [20]. It amounts to making move proposals according to

$$P(b_i^{(l)} = r \to s) = \sum_t P(t|i, l) \frac{e_{ts}^l + \epsilon}{e_t^l + \epsilon(B_l + 1)}, \tag{3.10}$$

[3]This constraint does not break ergodicity, since all partitions in the upper levels will be allowed to change at some point.

where $P(t|i,l) = \sum_j A_{ij}^{(l)} \delta(b_j^{(l)}, t)/k_i^{(l)}$ is the fraction of neighbors of node i in level l that belong to group t, and $\epsilon > 0$ is an arbitrary parameter that enforces ergodicity, but with no other significant impact in the algorithm, provided it is sufficiently small.

Furthermore, these proposals can be generated efficiently, simply by

1. sampling a random neighbor j of node i, and inspecting its group membership $t = b_j$, and then
2. with probability $\epsilon(B_l + 1)/(e_t + \epsilon(B_l + 1))$ sampling a fully random group s (which can be a new group),
3. or otherwise, sampling a group label s with a probability proportional to the number of edges leading to it from group t, e_{ts}.

While the above algorithm serves to sample from the posterior distribution of Eq. 3.5, it can be easily modified to find its maximum by introducing an "inverse-temperature" parameter β in Eq. 3.8 via the replacement $\Delta \ln P_l \to \beta \Delta \ln P_l$. By making $\beta \to \infty$ the algorithm is turned into a greedy heuristic that, if repeated many times, yields a reliable estimate of the maximum.

The division of the network into layers does not alter these algorithms in any significant way. We just need a book-keeping of the layer membership of each edge at each iteration. We refer to [19] for further details.

Remarks

The MCMC algorithm described in this chapter, for all model variants described, is implemented in the `graph-tool` *library* [21]*, freely available under the GPL license at* `http://graph-tool.skewed.de`*.*

Chapter

4

Bias and Variance Trade-Off

Summary

Given the posterior distribution $P(\boldsymbol{b}'|\boldsymbol{A})$, we can get estimates of inferred partitions in two ways. Firstly, we can maximize the posterior distribution Eqn. (2.3), which is equivalent to employing the MDL principle. Alternatively, we can collect a large number of samples after the Markov Chain has been equilibrated and then obtain the marginal distribution for each node, i.e. the probability that each node belongs to a given group. From this approach, we can obtain a "point estimate" for the partition by taking the maximum value for each node, i.e. the group membership with the largest probability. Following the reference [12], we devote this section by discussing the two estimators of partition. We will call the first as the maximum *a posteriori* (MAP) estimator and the second as the marginal estimator.

To see which estimator is more suitable, we need first define a *loss function* that compares the estimate $\hat{\boldsymbol{b}}$ of the partition to the true partition that generated the data $\boldsymbol{b}^*$. If we choose to be very strict, for example, we may reject any partition that is strictly different from $\boldsymbol{b}^*$ on equal measure, using the indicator function

$$\Delta(\hat{\boldsymbol{b}}, \boldsymbol{b}^*) = \prod_i \delta_{\hat{b}_i, b_i^*}, \tag{4.1}$$

so that $\Delta(\hat{\boldsymbol{b}}, \boldsymbol{b}^*) = 1$ only if $\hat{\boldsymbol{b}} = \boldsymbol{b}^*$, otherwise $\Delta(\hat{\boldsymbol{b}}, \boldsymbol{b}^*) = 0$. If the observed data $\boldsymbol{A}$ and parameters $\boldsymbol{b}$ are truly sampled from the model and priors, respectively, the best assessment we can make for $\boldsymbol{b}^*$ is given by the posterior distribution $P(\boldsymbol{b}|\boldsymbol{A})$. Therefore, the average of the indicator over the posterior is given by

$$\bar{\Delta}(\hat{\boldsymbol{b}}) = \sum_{\boldsymbol{b}} \Delta(\hat{\boldsymbol{b}}, \boldsymbol{b}) P(\boldsymbol{b}|\boldsymbol{A}). \tag{4.2}$$

4.1 The Maximum a Posteriori (MAP) Estimator of Partitions

If we maximize $\bar{\Delta}(\hat{\boldsymbol{b}})$ with respect to $\hat{\boldsymbol{b}}$, in Eqn. (4.2) we can obtain the MAP estimator

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b}}{\operatorname{argmax}}\, P(\boldsymbol{b}|\boldsymbol{A}). \tag{4.3}$$

However, using this estimator is arguably overly optimistic since the posterior distribution is complex and we are unlikely to find the true partition with perfect accuracy in most cases.

4.2 The Marginal Estimator of Partitions

Alternatively, we can consider the following overlap function

$$d(\hat{\boldsymbol{b}}, \boldsymbol{b}^*) = \frac{1}{N}\sum_i \delta_{\hat{b}_i, b_i^*}, \tag{4.4}$$

which measures the *fraction* of nodes that are correctly classified. If we maximize now the average of the overlap over the posterior distribution

$$\bar{d}(\hat{\boldsymbol{b}}) = \sum_{\boldsymbol{b}} d(\hat{\boldsymbol{b}}, \boldsymbol{b}) P(\boldsymbol{b}|\boldsymbol{A}), \tag{4.5}$$

we obtain the *marginal estimator*

$$\hat{b}_i = \operatorname*{argmax}_r \pi_i(r), \tag{4.6}$$

where

$$\pi_i(r) = \sum_{\boldsymbol{b}\setminus b_i} P(b_i = r, \boldsymbol{b} \setminus b_i|\boldsymbol{A}) \tag{4.7}$$

is the marginal distribution of the group membership of node i, summed over all remaining nodes.

Remarks

From the above, the marginal estimator is notably different from the MAP estimator since it leverages information from the entire posterior distribution to infer the partition that is responsible for the formation of the observed network. It is expected that

1. *If the posterior is tightly concentrated around its maximum, both estimators will yield compatible answers. In this situation the structure in the data is clear, and both estimators agree.*
2. *Otherwise, if the posterior possesses multiple peaks, the multiplicity of local maxima can be just a reflection of the randomness in the data, and the marginal estimator will be able to average over them and provide better accuracy* [33].

4.3 An Example Application

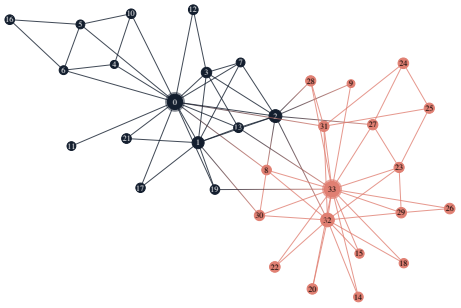
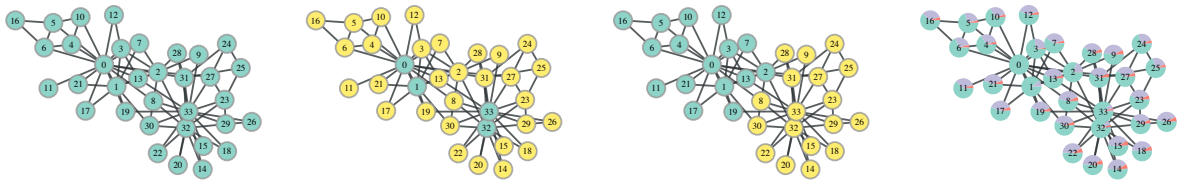


Figure 4.1 – *Zachary's karate club network.*

To illustrate the difference between the MAP estimator and the marginal estimator for the partition of a network, consider a well-known Zachary's karate club network [34] as a concrete example. As shown in Fig. 4.1, the 34 nodes in this network represent members of a karate club and a link between two members represent the interaction outside the club. The observed community structure[1] corresponds to a conflict between the administrator (node 0) and the instructor (node 33), ultimately leading to the split of the club into two factions. In the network science literature, this network serves as a common test to judge the quality of a community detection algorithm based on whether it can recover the observed partition.

[1] The observed partition of a network is often called "ground-truth" communities in the network science literature.

If we analyze this network with the DC-SBM, we can obtain the following three partitions with high posterior probabilities

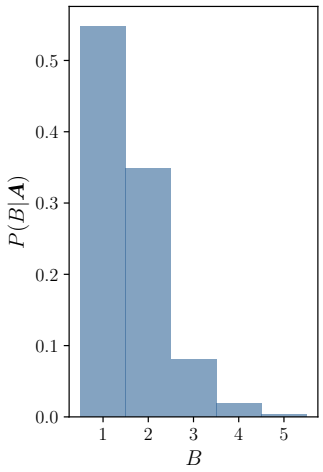


(a) – $\boldsymbol{b}_0$, $\Sigma = 323$ *bits*

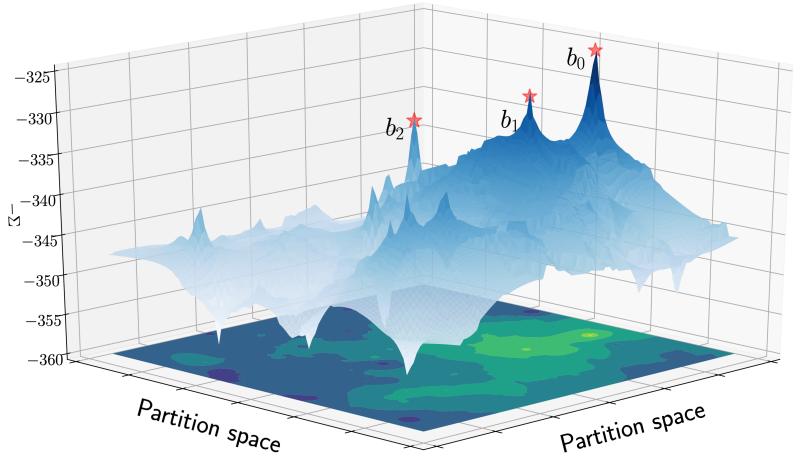
(b) – $\boldsymbol{b}_1$, $\Sigma = 327.5$ *bits.*

(c) – $\boldsymbol{b}_2$, $\Sigma = 329.3$ *bits*

(d) – *Marginal probabilities of group memberships.*



(e)



(f)

Figure 4.2 – ***Posterior distribution of partitions of Zachary's karate club network using the degree-corrected SBM.*** *Panels (a) to (c) show three modes of the distribution and their respective description lengths (measured in bits); (d) Marginal probabilities of group memberships of the Zachary's karate club network, according to the degree-corrected SBM. The pie fractions on each node represent the probability of being in community associated with the respective color. (e) Marginal posterior distribution of the number of groups B; (f) 2D projection of the posterior distribution, visualized using multi-dimensional scaling method (see Appendix A.1.2).*

- $\boldsymbol{b_0}$ in Fig. 4.2a: a trivial $B = 1$ partition. This indicates that the karate club network has no community structures.
- $\boldsymbol{b_1}$ in Fig. 4.2b: a "leader-follower" division into $B = 2$ fractions that separate the administrator and instructor together with two close allies from the rest of the network. This is an example of "core-periphery" structure that can be captured by SBM as described in Chapter 2.
- $\boldsymbol{b_2}$ in Fig. 4.2c: a $B = 2$ division into the groups that match the observed communities exactly. This is an example that SBM discovers the assortative structures in the network.

Since $\boldsymbol{b_0}$ occurs with highest posterior probability, this MAP estimator indicates the most likely explanation of this network is a fully random graph. On the other hand, if we obtain the marginal probabilities for the block memberships of each node as shown in Fig. 4.2d, we conclude that the marginal estimator for the partition of this network is still the trivial partition. Therefore, the over-reliance on this network to judge the quality of community detection methods is highly questionable.

As depicted in Fig. 4.2e, however, if we inspect the posterior distribution more closely, the sum of the posterior probabilities of other partitions into $B > 1$ groups is approximately equal to 0.5. Hence, if we consider all $B > 1$ partitions collectively, we cannot completely discard the possibility that the network posses some group structure. As shown in Fig. 4.2f, the posterior distribution of partitions reveals a multimodal structure clustered around the above three partitions. Hence, each of them might be a possible explanation for the Zachary's karate club network.

4.4 Discussion

The scenarios encountered for the karate club network illustrates the problem of *bias-variance tradeoff* in statistics and machine learning:

- **More bias, less variance.**
  If we choose to use a singe partition as a unique representation of the network, we must invariably *bias* our result toward any of the above three most likely partitions, discarding the remaining ones at some loss of useful information.
- **Less bias, more variance.**
  Otherwise, if we choose to eliminate the bias by incorporating the entire posterior distribution in our representation, it will incorporate a larger variance. That is, it will simultaneously encompass diverging explanations of the data, leaving us without an unambiguous and clear interpretation.

As discussed above, the only situation where this trade-off is not required is when the model is a perfect fit to the data, such that the posterior is tightly peaked around a single partition.

In view of the above, it might be argued that the marginal estimator for the partition should be generally preferred over MAP estimator. However, the situation is more complicated when the model is *misspecified*, that is, we use the model to fit the data that is in fact not generated from the assumed model. In this case, multiple peaks of the posterior distribution can point to very different but all statistically significant partitions. The partitions corresponding to these different peaks serve as alternative explanations for the data that must be accepted on same footing, according to their posterior probability. The marginal estimator will in general mix the properties of all peaks into a consensus classification that is not representative of any single hypothesis, whereas the MAP estimator will concentrate only on the most likely one.

The final decision on which approach to employ replies on the computational resources available and the actual objective. Generally, if the goal is to make a precise statement about the data, and the computational resources are limited, the MAP estimator tends to be more adequate. In contrary, when computational resources are ample, the marginal estimator will be more suitable if the objective is to generalize from observations and make predictions.

In this thesis, we will use both estimators for the partition of documents in all of the networks of interests when assessing the partition similarities between different models.

Chapter 5

Modeling Topicality

Summary

5.1 Connecting Topic Models and Community Detection

We devote this section to present the equivalence between the probabilistic latent semantic indexing (pLSI) from topic modelling and the stochastic block models from community detection, as shown in [28].

pLSI

Probabilistic Latent Semantic Indexing (pLSI) is defined as a generative process [23] that generates a corpus of D documents as follows.

- For each topic $r = 1, 2, \ldots, K$
  - Draw the word-topic distribution ϕ_w^r (frequencies of words conditioned on the topic r)
- For each document $d = 1, 2, \ldots, D$
  - Draw the topic-document distribution θ_{dr} (frequencies of topics conditioned on the doc d)
  - For each work-token $i_d = 1, 2, \ldots, k_d$ in document d
    - Draw a topic r_{i_d} from the categorical distribution θ_{dr}
    - Draw a word-type w_{i_d} from the categorical distribution $\phi_{r_{i_d} w}$

If we assume that the number of words k_d in a document d is Poisson-distributed with parameter η_d, and denote n_{dw}^r as the number of frequencies of word w of topic r in document d, the probability of generating a corpus composed of D documents is

$$P(\boldsymbol{n}|\boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_d \eta_d^{k_d} e^{-\eta_d} \prod_{wr} \frac{(\phi_{rw}\theta_{dr})^{n_{dw}^r}}{n_{dw}^r!}. \tag{5.1}$$

We denote matrices by bold-face symbols, e.g. $\boldsymbol{\theta} = \{\theta_d^r\}$ with $d = 1, \ldots, D$ and $r = 1, \ldots, K$ where θ_{dr} is an individual entry, thus the notation $\boldsymbol{\theta}_d$ refers to the vector $\{\theta_{dr}\}$ with fixed d and $r = 1, \ldots, K$.

Another SBM variant: Group overlaps

Another way we can change the internal structure of the SBM is to allow the groups to overlap. That is, a node can belong to more than one group simultaneously. Following the reference [27], we can write the likelihood function for SBMs with overlapping groups as

$$P(\boldsymbol{A}|\boldsymbol{\kappa}, \boldsymbol{\lambda}) = \prod_{i<j} \frac{e^{-\lambda_{ij}} \lambda_{ij}^{A_{ij}}}{A_{ij}!} \prod_i \frac{e^{-\lambda_{ii}/2} (\lambda_{ii}/2)^{A_{ii}/2}}{A_{ii}/2!}, \tag{5.2}$$

with

$$\lambda_{ij} = \sum_{rs} \kappa_{ir} \omega_{rs} \kappa_{js}, \tag{5.3}$$

where κ_{ir} is the probability that node i is sampled from group r and λ_{rs} is the expected number of edges between group r and group s.

The Equivalence between pLSI and SBM with group overlaps

To show the equivalence between the pLSI and the SBM with overlapping groups, we write the token probabilities in Eqn. (5.1) in a symmetric fashion as

$$\phi_{rw}\theta_{dr} = \eta_w \theta_{dr} \phi'_{wr}, \tag{5.4}$$

where $\phi'_{wr} \equiv \phi_{rw} / \sum_s \phi_{sw}$ is the probability that the word w belongs to topic r, and $\eta_w \equiv \sum_s \phi_{sw}$ is the overall propensity with which the word w is chosen across all topics. Hence, the likelihood of Eqn. (5.1) becomes

$$P(\boldsymbol{n}|\boldsymbol{\eta}, \boldsymbol{\phi}', \boldsymbol{\theta}) = \prod_{dwr} \frac{e^{-\lambda^r_{dw}} (\lambda^r_{dw})^{n^r_{dw}}}{n^r_{dw}!}, \tag{5.5}$$

with $\lambda^r_{dw} = \eta_d \eta_w \theta_{dr} \phi'_{wr}$. Further, if we choose to view the frequencies n_{dw} as the entries of the adjacency matrix of a bipartite network of documents and words, the likelihood of Eqn. (5.5) is equivalent to the likelihood of Eqn. (5.2) of the SBM, if we assume that each document belongs to its own specific group, $\kappa_{ir} = \delta_{ir}$, with $i = 1, \ldots, D$ for document-nodes, and by re-writing $\lambda^r_{dw} = \omega_{dr}\kappa_{rw}$.

Remarks

From the above, it is concluded that the SBM of Eqn. (5.2) is a generalization of pLSI. One advantage of this formulation is that it allows to not only cluster the words into topics but also to cluster the documents into groups. So pLSI is a special case of when the documents are not clustered.

5.2 Parallelism Between Topic Models and Community Detection Methods

Latent Dirichlet Allocation (LDA)

For an unknown text corpus, if we assume it is generated by the pLSI, we can simply maximize the likelihood Eqn. (5.1) to obtain the best estimators $\boldsymbol{\eta}$, $\boldsymbol{\theta}$, and $\boldsymbol{\phi}$, which describe the topical structure of the corpus. However, this maximum likelihood estimators will invariably incorporate a considerable amount of noise since the number of parameters in the model grow with the number of documents, words, and topics. To lessen the issue of overfitting, the approach proposed by [26] is employed by putting Dirichlet prior distributions $D_d(\boldsymbol{\theta}_d|\boldsymbol{\alpha}_d)$ and $D_r(\boldsymbol{\phi}_r|\boldsymbol{\beta}_r)$ with hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for the probabilities $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ above.

Modelling Documents and Words Using the hSBM

If we view the text corpus as a bipartite network generated by the SBM, we can use the nonparametric Bayesian approach that models the network as a hierarchy of SBMs in Chapter 2.

Since the bipartite network of documents and words, where they belong to different groups, is a special case of an arbitrary multigraphs generated by the hSBM, we can use the model as it is, as it will "learn" the bipartite structure during inference. However, a more consistent approach for text is to include this information in the prior, since we should not have to infer what we already know. Hence, we simply modify the model by replacing the prior for the partition at each level of the hierarchy by

$$P\left(\boldsymbol{b}_l\right) = P_w\left(\boldsymbol{b}_l^w\right) P_d\left(\boldsymbol{b}_l^d\right) \tag{5.6}$$

In this manner, by construction, words and documents will never be placed together in the same group.

Comparison between LDA and hSBM to topic modeling

We devote this section to summarize the advantages of the network approach (hSBM) over LDA, as shown in [28].

- Since the hSBM is based on nonparametric Bayesian inference, the number of topics is discovered automatically.
- Since the priors are hierarchical, the thematic structures can be discovered on many scales of resolution. In addition, the documents themselves can also clustered into hierarchical categories.
- If we perform both methods on artificial corpora sampled from LDA, the hSBM performs better in terms of description lengths. The improvement of the hSBM over LDA in a LDA-generated corpus is counterintuitive because, for sufficient data, we expect the true model to provide a better description for it.

Summary

Now, we have discussed the two different approaches – the traditional topic modeling technique (pLSI and LDA) and the network approach (SBM and hSBM) – to solve the problem of topic modeling. We summarize their relationship using Fig. 5.1.

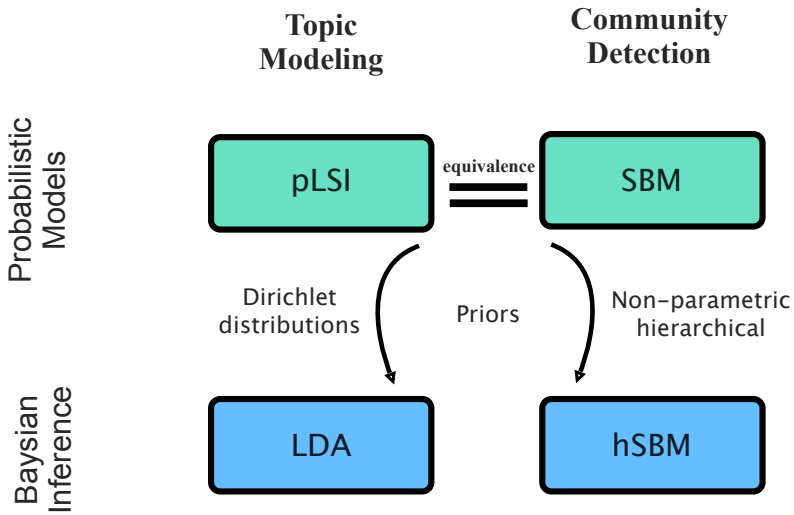


Figure 5.1 – ***Parallelism between topic models and community detection methods.*** *The pLSI and SBM are mathematically equivalent as shown in Sec. 5.1 and thus methods from community detection (described in Chapter 1) can be used as alternatives to traditional topic models (LDA).*

5.3 Modelling Texts With Auxiliary Information Using the SBM with Independent Layers

In this section, we summarize the two important assumptions in modeling the texts with auxiliary information by the SBM with independent layers as described in Sec. 2.4.

- **Independent layers:** The word tokens, hyperlinks between documents, and metadata tags are different types of interactions. We assume that there are no relationships among these different types of interactions.
- **Degree correction:** One document that receive many words does not necessarily mean that it will possess many hyperlinks or metadata tags.

5.4 Evaluations on Topic Models: Document Clustering

As mentioned above, one of the advantages of the network approach to topic modeling is that we can obtain the clustering of documents automatically. Given two partitions of documents, a natural question arises of how to compare the similarity between the two clusterings. The *normalized mutual information* (NMI) is such an information theoretic measure used for clustering comparison [37].

Let S be a set of N labels, then a clustering U on S is a way of partitioning S into non-overlap subsets $\{U_1, U_2, \ldots, U_R\}$, where $\cup_{i=1}^{R} U_i = S$ and $U_i \cap U_j = \emptyset$ for $i \neq j$. The information for the overlap between two partitions $\mathbf{U} = \{U_1, U_2, \ldots, U_R\}$ and $\mathbf{V} = \{V_1, V_2, \ldots, V_C\}$ can be summarized in the $R \times C$ *contingency table* $M = \left[n_{ij}\right]_{j=1,\ldots,C}^{i=1,\ldots,R}$, as shown in 5.1, where n_{ij} is the number of objects that are in common for clusters U_i and V_j.

$\mathbf{U}\backslash\mathbf{V}$	V_1	V_2	$\ldots$	V_C	Sums
U_1	n_{11}	n_{12}	$\ldots$	n_{1C}	a_1
U_2	n_{21}	n_{22}	$\ldots$	n_{2C}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_R	n_{R1}	n_{R2}	$\ldots$	n_{RC}	a_R
Sums	b_1	b_2	$\ldots$	b_C	$\sum_{ij} n_{ij} = N$

Table 5.1 – *The contingency table of two partitions of a set with N elements, where* $n_{ij} = \left|U_i \cap V_j\right|$
.

If we consider a partition as a distribution (probability of one node falling into one community), then given two partitions U and V, their entropies $H(\mathbf{U})$, joint entropy $H(\mathbf{U}, \mathbf{V})$, conditional entropies $H(\mathbf{U}|\mathbf{V})$ and mutual information (MI) $I(\mathbf{U}, \mathbf{V})$ can be calculated naturally using the following formulas

$$H(\mathbf{U}) = -\sum_{i=1}^{R} \frac{a_i}{N} \log \frac{a_i}{N}, \tag{5.7}$$

$$H(\mathbf{U}, \mathbf{V}) = -\sum_{i=1}^{R} \sum_{j=1}^{C} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N}, \tag{5.8}$$

$$H(\mathbf{U}|\mathbf{V}) = -\sum_{i=1}^{R} \sum_{j=1}^{C} \frac{n_{ij}}{N} \log \frac{n_{ij}/N}{b_j/N}, \tag{5.9}$$

$$I(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{R}\sum_{j=1}^{C}\frac{n_{ij}}{N}\log\frac{n_{ij}/N}{a_i b_j/N^2}. \tag{5.10}$$

It is noted that the MI is to quantify how much knowing $\mathbf{V}$ reduces the uncertainty about $\mathbf{V}$, and vice versa. Then, the NMI is defined as

$$\mathrm{NMI}_{sum} = \frac{2I(\mathbf{U},\mathbf{V})}{H(\mathbf{U}) + H(\mathbf{V})} \tag{5.11}$$

The NMI is normalized in the sense that the it is bounded between the minimum 0 when the labels are totally different in U and V, and the maximum 1 when U and V are identical. It is pointed out that there are several other variants of the NMI measure if we choose the different normalized constants appeared in the denominator of Eqn. (5.11). However, if we want compare partition similarities between different observed clusterings, the ranking of similarity values will not be affected by these choices of NMI measures.

Chapter 6

Experiments and Results

Summary

This chapter summarizes the numerical results of making use of auxiliary information that is available on Wikipedia for the application of the SBM framework to Wikipedia articles. In particular, we incorporate the hyperlinks, category labels, and the text content. There are three main results present in this thesis. Firstly, we incorporate texts with additional information about documents as multilayered network and visualize the inferred result in an informative way. Then, we perform a convergence analysis of the MCMC algorithms. Finally, we conclude this chapter by assessing whether incorporating more information about documents can help improve the their classification.

6.1 Case Study: The Wikipedia Articles

The Wikipedia is a multilingual online encyclopedia edited by volunteers. In this thesis, we only consider the English version of the Wikipedia. It is noted that, since the Wikipedia is dynamic, the frozen version of the English Wikipedia dump at 1st April, 2019 is used for reproducible research.

6.1.1 Text Preprocessing for Wikipedia Articles

We devote this section briefly describe all the data processing steps we took to obtain the corpus from the raw data (Fig. 6.1). The processing yields data for articles on 5 different levels of granularity:

- *Raw* data: We download the whole English Wikipedia.
- *Raw* data (subset): We further extract the articles based on its category labels.
- *Cleaned* Text: We remove all unwanted characters, e.g. LaTeXcodes, punctuations.

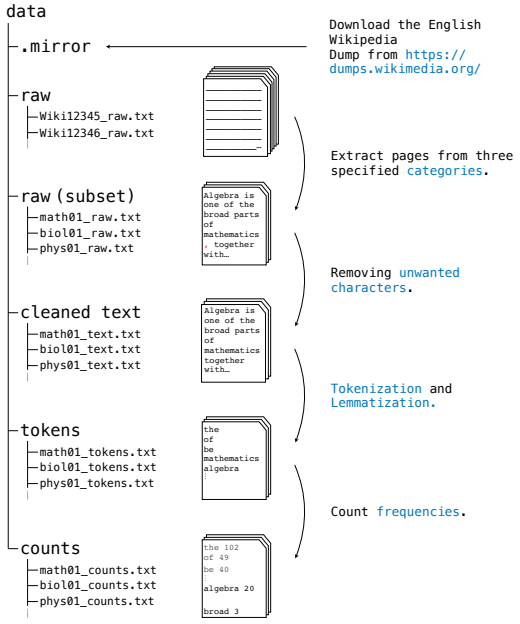


Figure 6.1 – ***Sketch of the preprocessing pipeline for the Wikipedia articles.*** *The folder structure (left) organizes each Wikipedia article on 5 different levels of granularity, see example article (middle): raw, raw texts with specified categories, cleaned text, tokens, and counts.*

- *Token* data: We tokenize[1] and lemmatize[2] the text data using the NLTK package [25].
- *Count* data: We count the number of occurrences of each word-type. This yields a list of tuples (w, n_w), where w is the word type and n_w is the number of occurrences.

Remarks

When subsetting the data in the above pipeline, we use an additional important filtering. That is, we exclude the Wikipedia article served as a navigation page to other articles. For example, we exclude the page with the title being "List of mathematical functions".

6.1.2 Datasets Summary

The above preprocessing pipeline provides a generic approach of how to extract articles from categories of our interests. In this Chapter, we consider articles from three categories, physics, mathematics and biology.

The observed hyperlink network for documents from three categories is shown in Fig. 6.2. It is shown that the hyperlink network exhibits the prescribed community structure, with a few exceptions. For instance, the yellow node colored with a red circle represents a Wikipedia page titled "computational anatomy" from the biology category. It is expected that, if we perform community detection on this hyperlink network, the algorithm is not supposed to categorizes as a biology article since it has more links connected to mathematics category. This is motivates the question how informative the hyperlinks between documents can improve the clustering of documents.

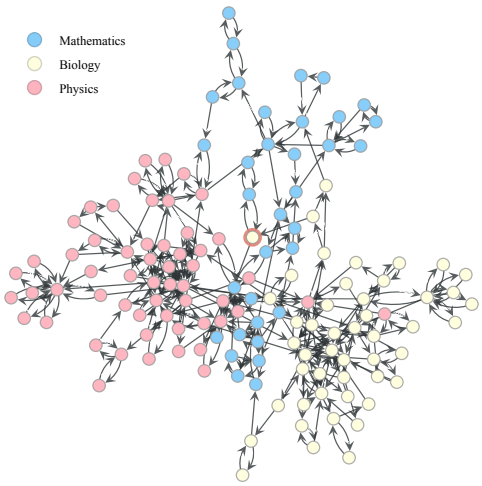


Figure 6.2 – *The observed hyperlink network $\mathcal{D}_1$ for the articles considered in this chapter.*

		$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$
Number of nodes	Document nodes	138	138	138	138	138
	Word types	-	16,378	16,378	16,378	16,378
	Labels	-	-	-	138	138
Number of edges	Word tokens	-	351,710	351,710	351,710	351,710
	Hyperlinks	341	-	341	-	341
	Metadata tags	-	-	-	138	138

Table 6.1 – *Summary of the datasets used in this chapter, showing the number of document nodes, word nodes (unique word tokens), tag nodes, word tokens, hyperlinks and tag edges.*

Based on the auxiliary information that is available on Wikipedia, we can construct five different networks incorporating different amount information available of documents, as illustrated in Fig. 6.3.

[1]Tokenization: split the text into tokens (words in our case).
[2]Lemmatization: convert the words into a common base form.

Additionally, as detailed in Table 6.1, we summarize the datasets considered in this Chapter by showing the number of documents, distinct words, category nodes, word tokens, hyperlinks, and metadata tags across different networks. The average text length is around 2,500 words.

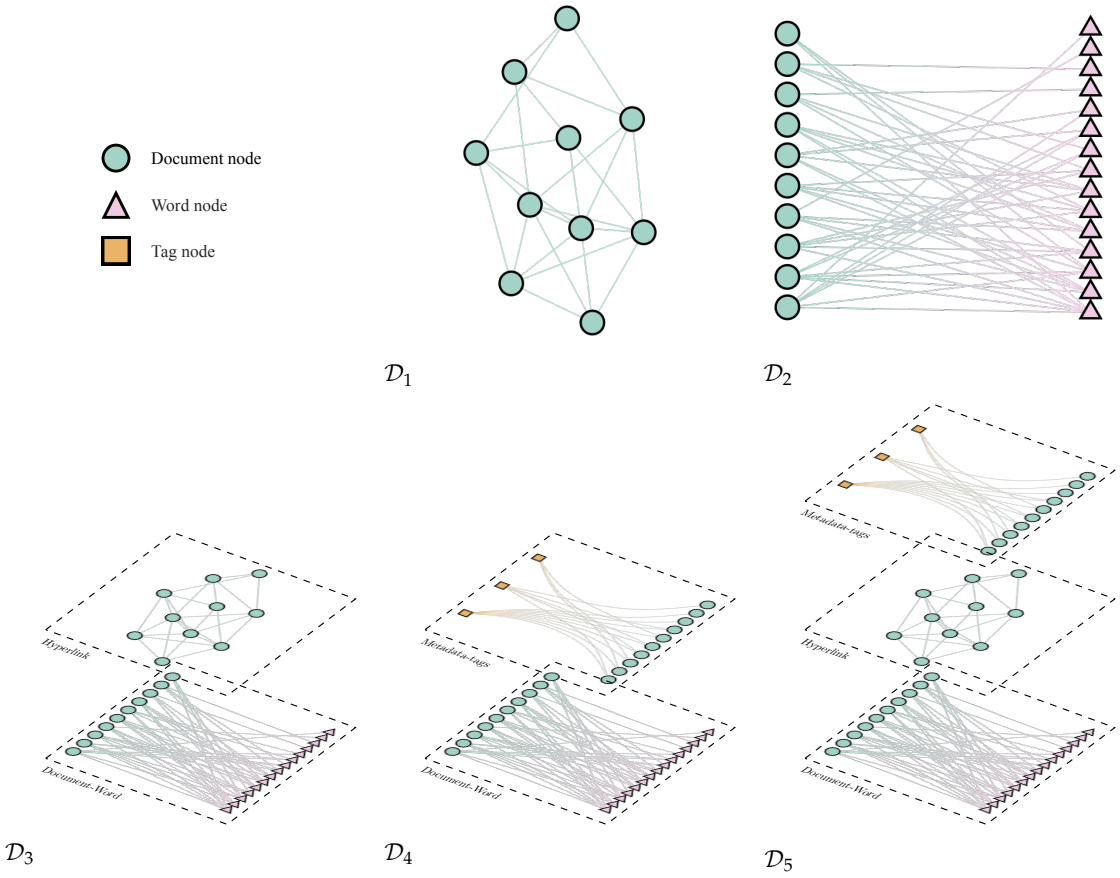


Figure 6.3 – ***The five possible networks we can construct from the Wikipedia datasets.*** *$\mathcal{D}_1$: The hyperlink network, where a node represents a document and an edges represents the hyperlink between two documents. $\mathcal{D}_2$: The bipartite network of documents and words, where the edges are word tokens. $\mathcal{D}_3$: The multilayered network, where the first layer is the document-word network and second layer is the hyperlink network. $\mathcal{D}_4$: The multilayer network where we incorporate the category labels about documents in the additional layer. $\mathcal{D}_5$: The three-layer network incorporating all information available about documents.*

6.2 Visualization of the Inference Results

In this section, we perform the algorithm based on the agglomerative heuristics introduced in Sec. 3.3 in the multilayered network $\mathcal{D}_2$, where the first layer corresponds to the bipartite network of documents and nodes and second one is the hyperlink network between documents.

We visualize the inference results by showing the bipartite network of documents and words. For making the visualization clearer, we uniformly randomly plot 1,000 of two types of edges – hyperlinks and word tokens, as shown in Fig. 6.4. Fig. 6.4 shows the hierarchical clustering of documents and words. The model splits the network into groups on different levels, organized as a hierarchical tree.

On the lowest level of the hierarchy, there are 5 document groups. For words, the lowest level in the hierarchy splits nodes into 59 separate groups. We find that, for example, there are groups representing

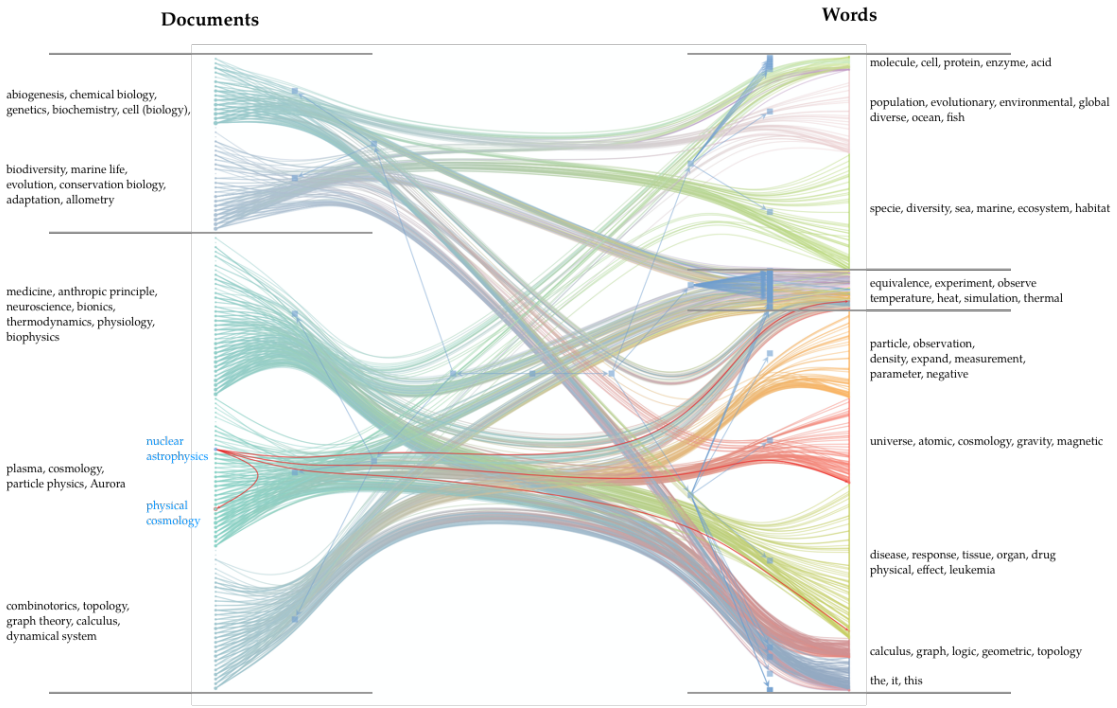


Figure 6.4 – ***The inference result from layered network*** $\mathcal{D}_3$**.** *Articles from three categories (mathematics, physics, and biology). The first hierarchical level reflects bipartite nature of the network with document nodes (left) and word nodes (right). The grouping on the second hierarchical level is indicated by solid lines. We show examples for nodes that belong to each group on this second hierarchical level: For word nodes, we show the most frequent words; for document nodes, we show five randomly selected articles.*

words belonging to biology (e.g. molecule, cell, and population) and mathematics (e.g. calculus, graph and logic), and the group representing function words (the, it, or this).

While we considered articles from three different categories, the *second* level in the hierarchy separates documents into only two groups corresponding to articles about biology and articles on physics and mathematics. For words, we summarizes the three topics (word groups) with the most frequent words in Table 6.2.

Topics	Top Words
Topic 1	the, of, in, a, to, be, and, for, that, it
Topic 2	field, system, time, number, state, theory, model, science, physic, problem
Topic 3	biology, life, size, human, evolution, biological, chemical, specie, environment, project

Table 6.2 – *Topical analysis of the learned hSBM on the* $\mathcal{D}_3$ *network, which displays the three topics with the top words at the second level of the hierarchy shown in Fig. 6.4.*

In summary, the model enables the identification of structural patterns in text, allowing for the identification of patterns in multiple scales of resolution for both documents and words.

6.3 Convergence Analysis of the MCMC Algorithms

The MCMC algorithm based on the agglomerative heuristics above for the Bayesian inference of the parameters is stochastic, and thus there is no guarantee that that two runs of the algorithm will yield the same result. As discussed in Chapter 5, this may be due to the fact that there are alternative partitions with similar probabilities, or that the optimum is difficult to find. Because of this, the approach Gerlach, Peixoto, and Altmann [28] adopted is to run the algorithm based on the agglomerative heuristics many times, and select the partition with the minimum description length.

In order to assess partition similarity of documents among different networks, this motivates a more careful analysis of the computational cost and convergence of the SBM fit. To this end, for each network considered, we compute the description lengths obtained after a growing number of MCMC iterations for different random choices of initial states. As discussed in Chapter 2, as long as the chain is ergodic, it will eventually converge to the desired posterior distribution. However, starting from a random partition may not be the best option, since it may take a long time for the chain to equilibrate. Hence, the approach we adopt in this thesis is to obtain multiple initial states by independently running the algorithm based on agglomerative heuristics multiple times. As such, the algorithm based on agglomerative heuristic is used as a privileged starting point for the Markov chain instead of as an approximate inference tool on its own [13].

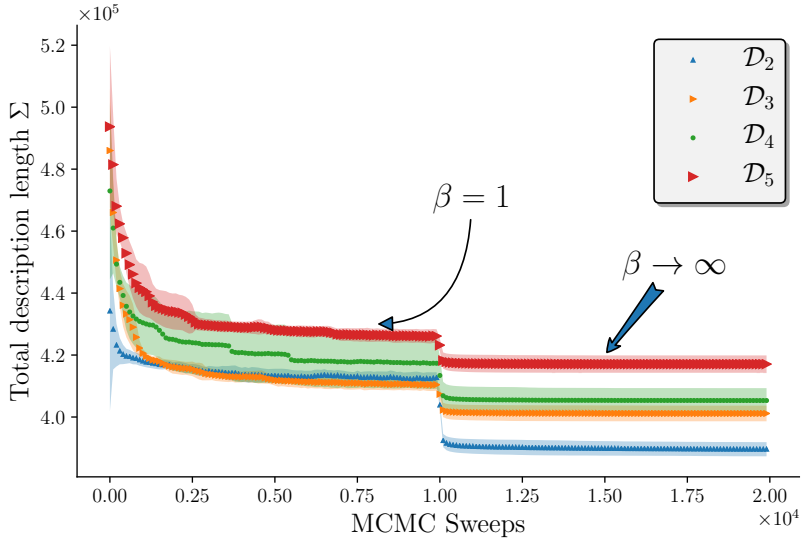


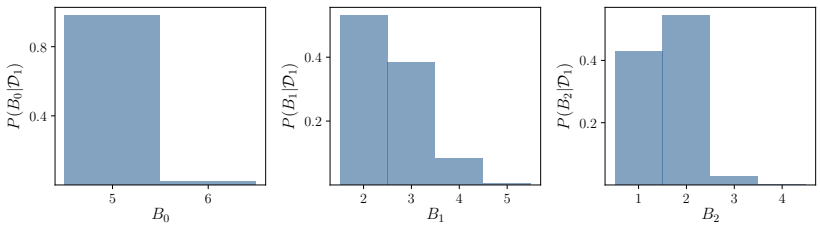
Figure 6.5 – ***The convergence of total description lengths for different networks.*** *For each network, the annealing was performed at* 10^4 *sweeps by switching the inverse temperature from* $\beta = 1$ *to* $\beta \to \infty$*. During each sweep, a move attempt is made for each node.*

After obtaining such starting states, we perform the MCMC algorithm for additional large enough iterations to see whether the description lengths converge or not. Specifically, for each run of the algorithm, we first sufficiently equilibrate the chain at the inverse temperature parameter $\beta = 1$, followed by abruptly cooling the chain via $\beta \to \infty$. Then, we report the average and standard deviation of description lengths for multiple runs over the MCMC iterations as shown in Fig. 6.5. This shows that the description lengths decay with the MCMC iterations. Importantly, there is a drastic decrease after the abrupt cooling of

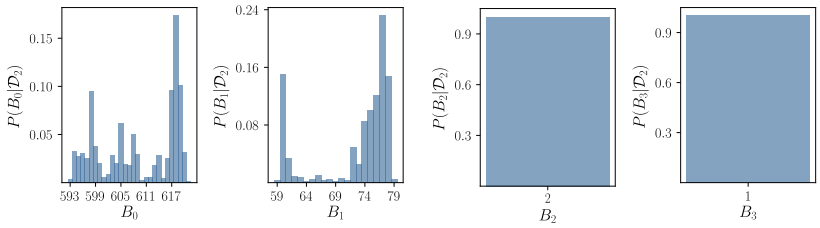
the chain, indicating that the annealing is important if we want to find the partition that minimizes the description length. In addition, although the standard deviations are also reduced, there is still some fluctuations in the end. This confirms that there exist alternative partitions with similar probabilities. Finally, for each run, we pick the state with the minimum description lengths observed throughout the full run.

The convergence analysis of the MCMC algorithm tells us that we should be open to the possibility that there will be more than one fit of the SBM with similar posterior probabilities. In such situations, we should instead sample partitions from the posterior distribution, instead of simply finding its maximum. We can then compute quantities that are averaged over the different model fits, weighted according to their posterior probabilities.

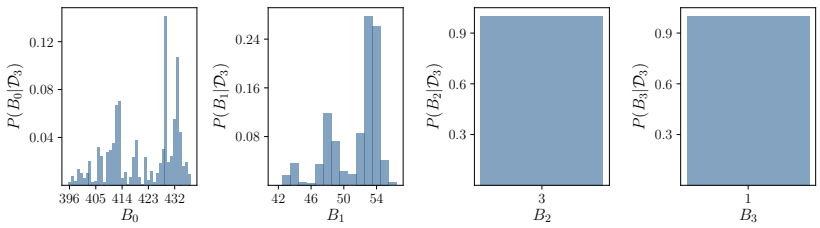
For example, we can obtain the marginal distribution of the number of groups over hierarchies as displayed in Fig. 6.7.



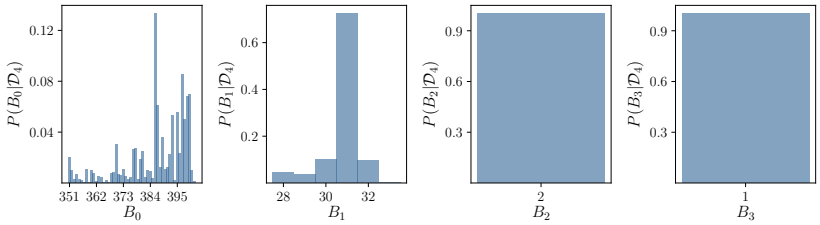
(a) – $\mathcal{D}_1$



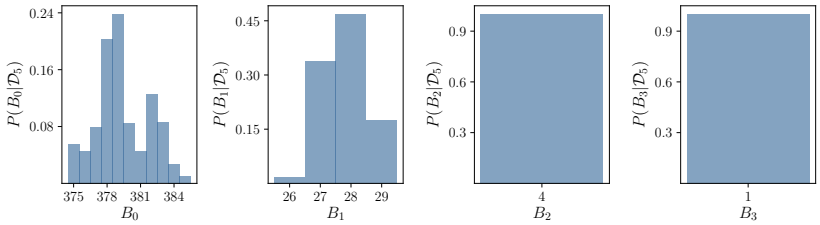
(b) – $\mathcal{D}_2$



(c) – $\mathcal{D}_3$



(d) – $\mathcal{D}_4$



(e) – $\mathcal{D}_5$

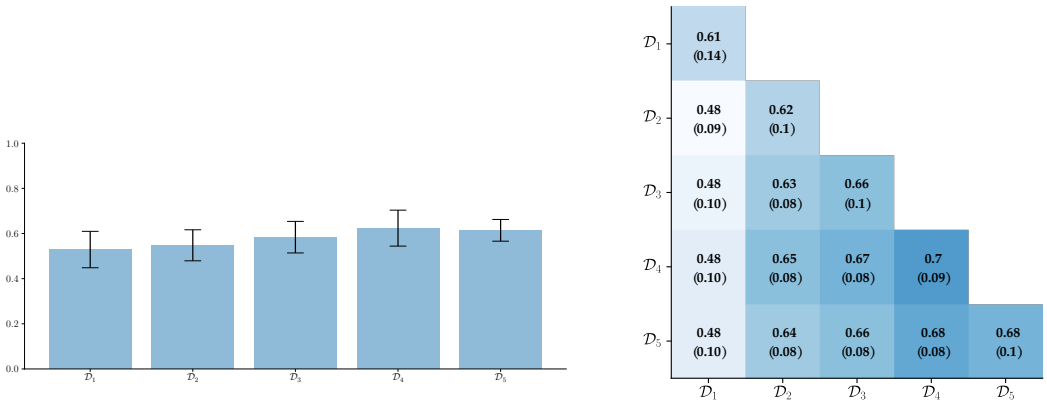
Figure 6.7 – *The marginal distribution of the number of groups B_j over the hierarchies j (left to right, $j = 0, \ldots, 3$) for different networks $\mathcal{D}_1, \ldots, \mathcal{D}_5$ (top to down).*

It is observed that the number of groups decrease drastically as we incorporate more information about the documents, indicating that adding more information in the models leads to a more definite answer to the network structures.

6.4 Comparison of Partition Similarities of Documents

6.4.1 Document Clustering I: Independent Runs of the MCMC Algorithms

After 20 independent runs of the MCMC algorithms for each dataset, we can obtain the partition in the region $\beta \to \infty$ shown in Fig. 6.5 and then compare the partition similarities of documents *within* and *between* different models. For that, we use the first level of the hierarchy (i.e., the one with the minimum number of documents), compute the NMI between the two different partitions, and then calculate the average and standard deviations of NMI values as shown in Fig. 6.8.



(a) – ***The average and standard deviation of NMI values compared to the Wikipedia labels.***

(b) – *The average and standard deviations (in brackets) of NMI values between and within different models.*

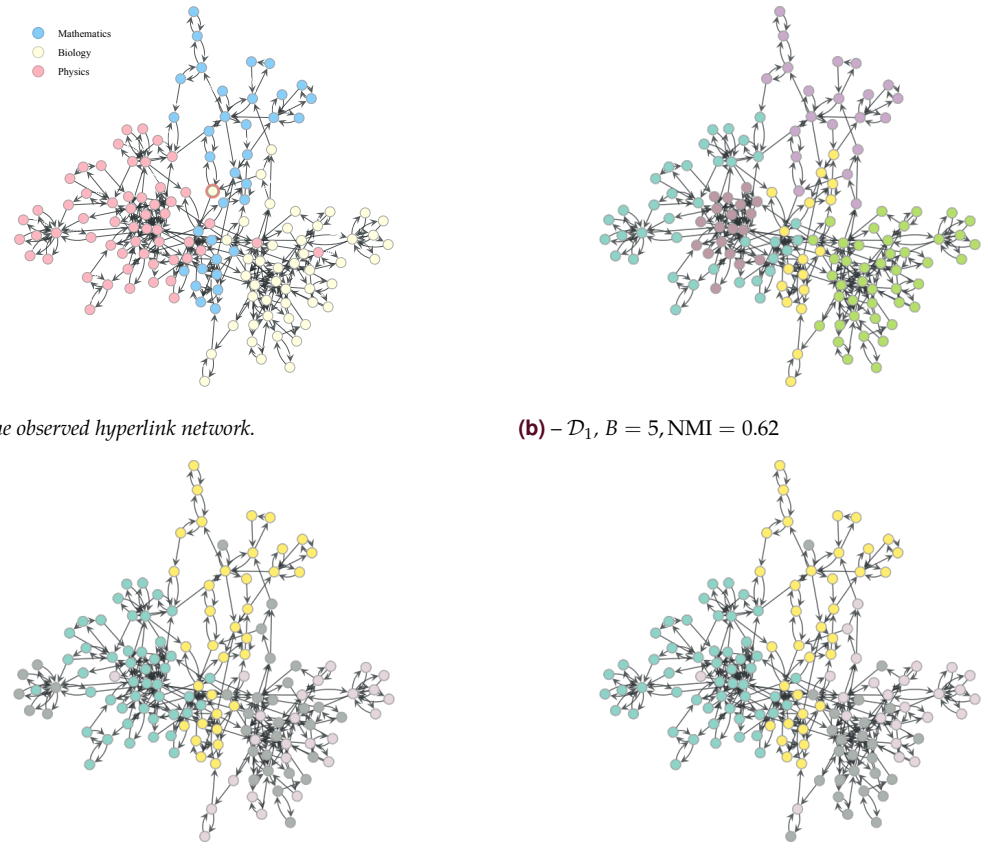
Figure 6.8 – *The partition similarities of documents.*

In Fig. 6.8a, compared to the Wikipedia labels, there is a slight increase of the NMI values as we incorporate more information in the model. And if we incorporate all the information available in the model, it is observed that the fluctuations also decrease.

As shown in Fig. 6.8b, the variation of the NMI within each model (i.e., how similar are two runs of the same model) is included in the diagonal of these tables. The values are comparable to the other values of the table, indicating that the variation of the different partitions found by the same model show a comparable diversity of results. Additionally, the models that contain words are more similar with each other than the model only containing the documents. Hence, the word tokens are dominating in the inference.

6.4.2 Document Clustering II: The Marginal Estimator of Partitions

Another possible estimate for the partition can be obtained via the Bayesian model averaging as discussed in Chapter 4. Specifically, after equilibrating the chains sufficiently, we can then collect a large number of samples from the posterior distribution in each model and obtaining the marginal distribution for each node, i.e. the probability that each node belongs to a given group. From this, we can obtain a "point estimate" for the partition by taking the maximum value for each node, i.e. the group membership with the largest probability.



(a) – *The observed hyperlink network.*

(b) – $\mathcal{D}_1$, $B = 5, \mathrm{NMI} = 0.62$

(c) – $\mathcal{D}_2$, $B = 5, \mathrm{NMI} = 0.67$

(d) – $\mathcal{D}_3$, $B = 4, \mathrm{NMI} = 0.71$

Figure 6.9 – ***The observed hyperlink network and the most likely inferred partition of documents for the networks*** $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$**.** *In the caption below each figure, we also include the number of inferred groups for the documents and NMI value compared to the Wikipedia labels.*

In this way, as discussed previously, the obtained partitions may not be the one that minimizes the description lengths. And the advantage of this approach is that it leverages the consensus over many partitions.

We compare the obtained partition with the Wikipedia labels as depicted in Fig. 6.9. We can see that in this case, there is also an increase in the NMI values, which confirms the conjecture that the more information we incorporate in the models, the more similar it is to the "ground-truth" labels.

Chapter 7

Concluding Remarks

Summary

7.1 Summary and Conclusion

The first part of the thesis introduces the main ideas of the problem of topic modeling, including traditional approaches, that represent the texts as a word-document matrix, and the network approach that considers the collection of documents as a network. We illustrate the task of topic modeling by a simple example and make a clear statement of the research problem and outline the structure of this thesis.

Chapter 2 summarizes the mathematical models (SBM and its variants) that are responsible for the formation of the networks containing words, documents, and auxiliary information about documents. Additionally, we describe the Bayesian inference of the SBM in SBM in detail in Chapter 2. Since the posterior distribution of the partition of networks is not simple enough to sample from or locating the maximum, we make a detailed literature review on the approximate inference method – Monte Carlo Markov Chain algorithm for different variants of SBMs in Chapter 3. In Chapter 4, we illustrate the problem bias-variance tradeoff in the estimation of the partition by using a small example. We showed that an advantage in Bayesian modeling is to perform model averaging as opposed to only considering the partition that maximizes the posterior distribution.

Then, Chapter 5 considers the first probabilistic topic model (pLSI) and shows its equivalence to the SBM formulation of document-word bipartite network. Then, we propose the new extended model incorporating the additional information available about documents using the SBM with independent layers.

Finally, Chapter 6 considers the Wikipedia articles as an example. Adopting an unsupervised learning approach, we find that incorporating more information leads to better agreement between the inferred partition of documents and the Wikipedia labelling. By comparing the results from detecting the communities on the hyperlink network of documents only, it is also observed that partitions of documents are more similar when incorporating words. Additionally, we also observe that the word tokens are dominating in the inference, since there are much more words than the number of hyperlinks or metadata tags.

7.2 Contributions

This thesis makes a number of contributions. Firstly, this project involved the development of an extensive software program, coded in `Python3`, for the purpose of conducting numerical simulations. We write all the code in functions or libraries where possible to facilitate future research. The topic modeling in text analysis and community detection in network science, by nature, intersects with the discipline of

computer science heavily, so computational analysis in `Python3` is an indispensable part of the research throughout the Honours year.

Secondly, we extend the previous work [28] by incorporating additional information available about documents as additional layers in the same SBM framework. We consider three cases with incorporating different amount of information present in the documents.

Thirdly, the previous research [28] does not perform a convergence analysis of the MCMC algorithm. In this thesis, we consider multiple independent runs of the algorithms starting from privileged initial states, which are obtained by the agglomerative heuristics. We find that the description lengths are substantially reduced with MCMC iterations and annealing, showing that those are essential steps in the inference of topic modeling with SBMs.

Last but not least, we compare the partition similarities of documents in two ways. Firstly, by comparing the final clustering of documents *within* and *between* different models. Secondly, we make use of model averaging by collecting a large number of samples from the posterior, and obtaining the group memberships with largest posterior probabilities.

7.3 Future Research

As future work, it would be interesting to apply the proposed topic models on other datasets. For example, we can test the algorithms on scientific papers, where the links represent citations. For research publications, additional auxiliary information that can be important includes the time of publication, the publication type, the conference venue, and the authors.

In this thesis, we only consider the metadata labels about the documents. It is also important to consider incorporating more information about words. Specifically, we can consider the word lexicons. For instance, we can make use of synonym and antonym lexicons for sentiment analysis.

Another interesting future research would be on comparing the extended models developed in this thesis with other topic models that make use of additional information for a complete analysis.

Instead of comparing the inferred partitions from each model with the "ground-truth" categories, a more interesting comparison would be in the context of link prediction reconstruction [18]. Specifically, we want to investigate whether adding word-document edges help in predicting missing/hidden hyperlinks.

Appendix

A

Summary

A.1 Technical Notes

A.1.1 The Prior Distribution for the Degree Distribution

Similarly to the partition of the nodes, the simplest choice we can make is to sample the degrees inside each group from a uniform distribution,

$$P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b}) = \prod_r \left(\!\!\binom{n_r}{e_r}\!\!\right)^{-1} \tag{A.1}$$

where $\left(\!\binom{n_r}{e_r}\!\right)$ counts the number of possible degree sequences on n_r nodes, constrained such that their total sum equals e_r. But again, such a uniform assumption is not the best choice: If we sample from this prior, we still obtain degree sequences where most nodes have very similar degrees.

In view of this, and following the same logic employed for the node partition, a better prior for $\boldsymbol{k}$ should be conditioned on an arbitrary degree distribution $\boldsymbol{\eta} = \{\eta_k^r\}$, with η_k^r being the number of nodes with degree k that belong to group r,

$$P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b}) = P(\boldsymbol{k}|\boldsymbol{\eta})P(\boldsymbol{\eta}|\boldsymbol{e},\boldsymbol{b}) \tag{A.2}$$

and where

$$P(\boldsymbol{k}|\boldsymbol{\eta}) = \prod_r \frac{\prod_k \eta_k^r!}{n_r!} \tag{A.3}$$

is a uniform distribution of degree sequences constrained by the overall degree counts, and

$$P(\boldsymbol{\eta}|\boldsymbol{e},\boldsymbol{b}) = \prod_r q(e_r, n_r)^{-1} \tag{A.4}$$

is the distribution of the overall degree counts. The quantity $q(m,n)$ is the number of different degree counts with the sum of degrees being exactly m and that have at most n non-zero counts.

We refer to [20] for further details of the above distribution.

A.1.2 Visualization Technique: The Multi-Dimensional Scaling Method

We devote this section to explain the methods used to produce the surface plot of description lengths as shown in Fig. 4.2, page 25. The surface plot of description lengths depicts that the changes in description length as the partition of network nodes varies. Although the surface appears to be continuous over that

two dimensional partition space, in spite of the fact that the true space of partitions is high dimensional and discretized. The steps for generating the surface plot is outlined as follows:

- **partitions sampling**
  It is infeasible to calculate the description lengths of all possible partitions for most networks, so we instead sample a subset of partitions. Suppose the number of nodes in the network of interest is N. A number of K partitions are sampled from the posterior distribution using MCMC after sufficient equilibration. And then store the K partitions with its corresponding description lengths.
- **partition similarity measure**
  Then, we calculate the partition similarities between the K partitions using variation of information (VI) [31]. We choose VI as a similarity measure since it is adequate for comparing partitions of different sizes. And the VI is a true metric since it preserves the distance between different partitions.
- **data projection**
  Next, the $K \times K$ partition similarity matrix is projected down to two dimensions using multi-dimensional scaling (MDS) [29]. The result of this projection is a two-dimensional representation of the partition space that preserves the VI between partitions.
- **surface interpolation**
  Since the two-dimensional partition space preserves the distance between different partitions of the network, we thus use the Nearest-neighbor interpolation technique to fit an interpolated surface to the description lengths of K partitions.

The last two steps are done by using the functions `manifold.MDS` and `neighbors.NearestNeighbors` respectively from the `scikit-learn` package [32] in `Python3`.

References

[1] Mark Newman. *Networks*. Oxford university press, 2018.

[2] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[3] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[4] Peter D Grünwald and Abhijit Grunwald. *The minimum description length principle*. MIT press, 2007.

[5] Jorma Rissanen. "Modeling by shortest data description". *Automatica* 14.5 (1978).

[6] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. "Stochastic blockmodels: First steps". *Social networks* 5.2 (1983).

[7] Brian Karrer and Mark EJ Newman. "Stochastic blockmodels and community structure in networks". *Physical review E* 83.1 (2011).

[8] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks". *Journal of statistical mechanics: theory and experiment* 2008.10 (2008).

[9] Mark EJ Newman. "Modularity and community structure in networks". *Proceedings of the national academy of sciences* 103.23 (2006).

[10] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. "Equation of state calculations by fast computing machines". *The journal of chemical physics* 21.6 (1953).

[11] W Keith Hastings. "Monte Carlo sampling methods using Markov chains and their applications" (1970).

[12] Tiago P Peixoto. "Bayesian stochastic blockmodeling". *arXiv preprint arXiv:1705.10225* (2017).

[13] Tiago P. Peixoto. "Efficient Monte Carlo and Greedy Heuristic for the Inference of Stochastic Block Models". *Physical Review E* 89.1 (2014).

[14] Tiago P. Peixoto. "Parsimonious Module Inference in Large Networks". *Physical Review Letters* 110.14 (2013).

[15] Darko Hric, Tiago P. Peixoto, and Santo Fortunato. "Network Structure, Metadata, and the Prediction of Missing Nodes and Annotations". *Physical Review X* 6.3 (2016).

[16] Tiago P. Peixoto. "Hierarchical Block Structures and High-Resolution Model Selection in Large Networks". *Physical Review X* 4.1 (2014).

[17] Tiago P. Peixoto. "Entropy of Stochastic Blockmodel Ensembles". *Physical Review E* 85.5 (2012).

[18] Tiago P. Peixoto. "Model Selection and Hypothesis Testing for Large-Scale Network Models with Overlapping Groups". *Physical Review X* 5.1 (2015).

[19] Tiago P. Peixoto. "Inferring the Mesoscale Structure of Layered, Edge-Valued, and Time-Varying Networks". *Physical Review E* 92.4 (2015).

[20] Tiago P. Peixoto. "Nonparametric Bayesian Inference of the Microcanonical Stochastic Block Model". *Physical Review E* 95.1 (2017).

[21] Tiago P. Peixoto. “The graph-tool python library”. *figshare* (2014).

[22] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. “Indexing by latent semantic analysis”. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41.6 (1990).

[23] Thomas Hofmann. “Probabilistic latent semantic analysis”. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1999.

[24] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. “The author-topic model for authors and documents”. *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press. 2004.

[25] Edward Loper and Steven Bird. “NLTK: The Natural Language Toolkit”. *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP ’02. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002.

[26] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. *Journal of machine Learning research* 3.Jan (2003).

[27] Brian Ball, Brian Karrer, and Mark EJ Newman. “Efficient and principled method for detecting communities in networks”. *Physical Review E* 84.3 (2011).

[28] Martin Gerlach, Tiago P. Peixoto, and Eduardo G. Altmann. “A Network Approach to Topic Models”. en. *Science Advances* 4.7 (2018).

[29] Inger Borg and PJF Groenen. “Springer series in statistics”. *Modern multidimensional scaling: Theory and applications (2nd ed.). New York, NY, US: Springer Science+ Business Media* (2005).

[30] Santo Fortunato and Marc Barthelemy. “Resolution limit in community detection”. *Proceedings of the national academy of sciences* 104.1 (2007).

[31] Marina Meilă. “Comparing clusterings by the variation of information”. *Learning theory and kernel machines*. Springer, 2003.

[32] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python ”. *Journal of Machine Learning Research* 12 (2011).

[33] Lenka Zdeborová and Florent Krzakala. “Statistical physics of inference: Thresholds and algorithms”. *Advances in Physics* 65.5 (2016).

[34] Wayne W Zachary. “An information flow model for conflict and fission in small groups”. *Journal of anthropological research* 33.4 (1977).

[35] L. Ingber. “Simulated annealing: Practice versus theory”. *Mathematical and Computer Modelling* 18.11 (1993).

[36] Jack Kiefer. “Sequential minimax search for a maximum”. *Proceedings of the American mathematical society* 4.3 (1953).

[37] Nguyen Xuan Vinh, Julien Epps, and James Bailey. “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance”. *Journal of Machine Learning Research* 11.Oct (2010).